REAL TIME IMPLEMENTATION OF A MILITARY IMPULSE NOISE CLASSIFIER

by

Matthew Brandon Rhudy

BS, Mechanical Engineering, Pennsylvania State University, 2008

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science in Mechanical Engineering

University of Pittsburgh

2009

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Matthew Brandon Rhudy

It was defended on

November 24, 2009

and approved by

William W. Clark, Professor, Department of Mechanical Engineering and Materials Science

William S. Slaughter, Associate Professor, Department of Mechanical Engineering and Materials Science

Jeffery S. Vipperman, Associate Professor, Department of Mechanical Engineering and Materials Science

Thesis Advisor: Jeffery S. Vipperman, Associate Professor, Department of Mechanical Engineering and Materials Science

REAL TIME IMPLEMENTATION OF A MILITARY IMPULSE NOISE CLASSIFIER

Matthew Brandon Rhudy, MS

University of Pittsburgh, 2009

A real time military impulse classifier has been developed to distinguish between impulsive events, such as artillery fire, and non-impulsive events, such as wind or aircraft noise. The classifier operates using an artificial neural network (ANN) with four scalar metrics as inputs. This classifier has been installed into two prototype noise monitoring systems, which are capable of establishing an accurate record of impulse events. This record can be used to assist in processing noise complaints and damage claims. The system continually monitors sound levels with a microphone array and activates when the sound level exceeds a given threshold. Once activated, the system processes the data to determine the classification, as well as the approximate bearing of the event.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

AMNOR	Adaptive Microphone-Array System for Noise Reduction
ANN	Artificial Neural Network
APS	Applied Physical Sciences Corp.
ARM	Advanced RISC Machine
B&K	Brüel & Kjær
BAMAS	Bearing and Amplitude Measurement and Analysis System
c	Speed of Sound in Air
CF	Crest Factor
CPU	Central Processing Unit
CST	Central Standard Time
DAQ	Data Acquisition
DOA	Direction of Arrival
dB	Decibels
dBC	Decibels using C-Weighting
DFR	Decimated, Filtered, and Re-sampled
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
EST	Eastern Standard Time
E_i	Relative Signal Error

FFT	Fast Fourier Transform
FLOP	Floating Point Operation
FN	False Negative
FP	False Positive
IPP	Integrated Performance Primitives
LD	Larson Davis
L_{pk}	Peak Sound Level
m	Spectral Slope
MFLOP	Mega- Floating Point Operation (10^6 FLOP)
MLP	Multi-Layer Perceptron
NI	National Instruments
p	Sound Pressure
p_{corr}	Overall Correlated Sound Pressure
$\|p_{pk}\|$	Maximum Absolute Instantaneous Pressure Amplitude
p_{rms}	Root Mean Square of Sound Pressure
PSD	Power Spectral Density
RISC	Reduced Instruction Set Computer
R	Gas Constant
RMS	Root Mean Square
R_{xx}	Autocorrelation Function of x(t)
R_{xy}	Cross-Correlation Function
SBC	Single Board Computer
SEL	Sound Exposure Level

SLM	Sound Level Meter
SMBus	Smart Management Bus
SNR	Signal-to-Noise Ratio
T_k	Absolute Temperature
UPS	Uninterruptible Power Supply
VI	Virtual Instrument
W_i	Un-normalized Weight
w_i	Normalized Weight
WSE	Weighted Square Error
XML	Extensible Markup Language
γ	Ratio of Specific Heats
θ	Angle of Incidence
μ	Mean Acoustic Sound Pressure
ρ_m	Min Peak Correlation Coefficient
ρ_{xy}	Cross-Correlation Coefficient Function
σ	Standard Deviation
τ	Channel-to-Channel Time Delay
τ_{lead}	Leading Microphone Time Delay
τ_{max}	Maximum Possible Time Delay

ACKNOWLEDGEMENTS

This research was supported wholly by the U.S. Department of Defense, through the Strategic Environmental Research and Development Program (SERDP). Thanks to Ellis Leeder, Sergio Sergi, and Mel Soult for their assistance in collecting data. Thanks to Paul Peterson and his team for their assistance in installing the prototype systems. Thanks to Jeffrey Allanach, Justin Borodinsky, and others at Applied Physical Sciences, Corp. (APS) for their cooperation and assistance in the development and execution of the overall system. Also, thanks to Brian Bucci for his guidance and assistance throughout the development of the project.

1.0 INTRODUCTION

During training sessions at military installations, impulse noise is generated, which propagates to the surrounding communities, often causing public annoyance (Schomer and Neathammer, 1985). Military impulse noise is defined by weapons firing projectiles with diameters of 20 mm or greater, or any type of high explosive charges, such as hand grenades or landmines. In addition to audible disturbances, military impulses can also induce vibration in housing, causing issues with house rattling, structural damage, *etc*. (Buchta and Vos, 1998). Because of these annoyance problems there is a need to establish a method to monitor and record military impulse events in order to corroborate damage claims ("Joint Land Use Study", 1993, Luz, 1980, 2004).

To address this issue, monitoring systems have been put into place around military installations. These systems keep record of all noise events that exceed a defined noise level by continuously monitoring C-weighted noise levels, which approximate the human ear's response to a 90 phon loudness level (Norton, 2003). C-weighting is effective in estimating the human ear response for military impulse noise of high amplitude. When levels exceeded 115 dBC, sound exposure level (SEL), time, duration, and peak level of the event were recorded by the system. A 2nd-generation of monitoring stations was developed based on an algorithm that assumes an impulse event waveform looks similar to a classic Friedlander wave (ANSI, 1993) and complies with a set of temporal conditions. This system also requires that peak sound levels exceed 115 dBC in

order for an event to be registered. This reasonably high threshold is used because one of the major problems with current noise monitoring systems is that false events are being detected due to wind noise (Schomer, 1988). At lower threshold levels, more false positive events are reported due to wind. This monitoring system is reasonably effective when impulse events have levels of about 119 dBC (ISO, 1996) and conditions are ideal (Schomer and Attenborough, 2005, Chunchuzov *et al.*, 2005). A 3rd-generation noise monitoring system is also currently in use. This system uses a modified sonic boom detection algorithm as a means for reducing false positive events. These stations however still suffer from significant false positive detections, which are primarily due to wind (Bucci, 2007).

In order to help reduce the number of false detections, an algorithm is required to classify a given waveform as impulse or non-impulse noise, such as wind. In previous work by Bucci, various classification algorithms were investigated. One of the most successful methods was found to be an artificial neural network (ANN) method, with accuracies reaching up to 100% (Bucci, 2007). This ANN classifier was developed as a post-processing method only, based on field collected data. This classifier utilized a single channel of data to obtain a classification result. As an alternative to single channel work (Bucci and Vipperman, 2006, 2007, Bucci, 2007), multiple channel methods using a microphone array were developed in collaboration with Applied Physical Sciences Corp. (APS). The additional degrees of freedom provided by a microphone array allowed for a more detailed analysis of noise events (Rhudy *et al.*, 2009).

In order to effectively create an accurate noise record, it was desired to implement a military impulse noise classifier in real time. To accomplish this, implementation on a

digital signal processing (DSP) board was investigated. This thesis presents the development of the classifier, including the exploration of different multiple channel methods, as well as the development of the final real time implementation.

2.0 BACKGROUND INFORMATION

2.1 ARTIFICIAL NEURAL NETWORKS

An artificial neural network (ANN) is a computational structure that was designed to roughly simulate the nerve cell networks of the biological central nervous system (McCulloch and Pitts, 1943). The structure of an ANN consists of multiple artificial neurons, with weighted interconnections, called synaptic weights (Graupe, 1997). One particular ANN structure is the multi-layer perceptron (MLP), which consists of an input layer, an output layer, and hidden layers of neurons. In general, each neuron in a layer is connected to all of the neurons in the adjacent layers. These connections are one-way, and are represented by the synaptic weights (Cichocki and Unbehauen, 1993). **Figure 1** shows a diagram of a MLP, with m inputs (x_1, x_2, …, x_m) and n outputs (y_1, y_2, …, y_n).

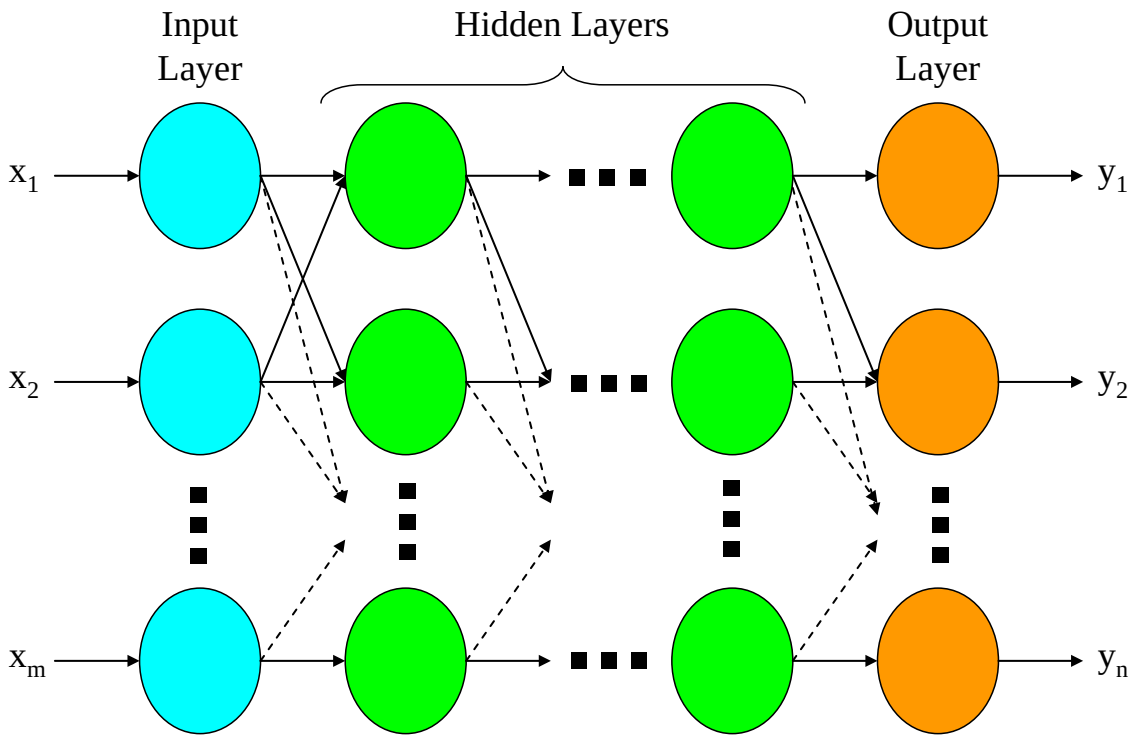

Figure 1. Diagram of Multi-Layer Perceptron (MLP) Structure

After creating the structure of an ANN, it must be trained in order to properly execute its desired task. There are two types of learning methods for training an ANN, supervised learning and unsupervised learning (Haykin 1999). Supervised learning involves comparing the output of the network with the desired output of the network for the given inputs. Unsupervised learning allows the network to adapt itself to the data space by identifying similar data points. Both of these methods involve adjusting the synaptic weights to obtain the desired result. An example of a supervised learning method is the Levenberg-Marquardt algorithm, which is an optimized method (Chong, 2001).

In order to train and evaluate an ANN network using supervised learning, a data set of input and output pairs is required. This data is used in three different ways to help

create the best possible network. The data set is divided into training, validation, and test data. The training data is used to adjust the synaptic weights by comparing actual output with desired output. This is the primary data set and is typically the largest portion of the data. The validation set is included in order to make sure the network is not being over-fitted to the training set. This means that the validation set is used to maintain the generality of the network, so that it can be applied to new data that was not used to train, validate, or test the network. Achieving maximum training accuracy is not necessarily the best case scenario, since the real goal is to obtain a classifier that will be most accurate for test data. In order to truly judge the performance of an ANN, the test data set is used. This test data has no effect on the training of the network, and is often called "blind," since it does not "see" any of the training aspects of the network. The test data is passed through the network to obtain an output, which is compared with the given desired output. The accuracy of the test data set can be found from this information, which is only used as a gauge of performance, and it is not used in any way to train the network further. The reason test data is used to measure performance is due to the fact that it represents the case of real world applications, where the desired output is not necessarily known.

2.2 MICROPHONE ARRAYS

An array is a set of sensor elements that are arranged at different points in space in order to obtain various effects. Sensor arrays have applications in fields such as radar, sonar, seismology, radio astronomy and tomography (Haykin, 1985). Microphone arrays are

used to investigate the different characteristics of acoustic signals as they propagate across each microphone. Three common applications of microphone arrays are noise rejection, beamforming, and tracking.

2.2.1 Noise Rejection

A common problem when dealing with signal measurement is contamination due to noise. Signal processing techniques have been developed to deal with this issue. In particular, sensor arrays can be used to detect the desired signal while reducing the effects of undesired noise. In order to detect whether a signal is present in a measurement, the signal must be distinguished from the noise by some property. For array problems, the simplest case of this is that the signal is propagating. When looking at array processing problems, it is typically assumed that the waveform $x_i(t)$ at each of M sensors contains both a signal, which is common to all of the sensors, and random noise, which is statistically independent from sensor to sensor. This is represented as

$$x_i(t) = s(t) + n_i(t) \tag{1}$$

where $s(t)$ is the signal and $n_i(t)$ is the noise at the ith sensor. By taking the average over the M signals, the effects of noise can be reduced.

$$y(t) = \frac{1}{M}\sum_{i=1}^{M} x_i(t) = s(t) + \frac{1}{M}\sum_{i=1}^{M} n_i(t) \tag{2}$$

It can be seen from equation (2) that the output of the averaging will contain the desired signal added to the average of the random noise components. These noise components will tend to cancel out due to their uncorrelated and random nature, thus leaving only the desired signal (Johnson and Dudgeon, 1993). To help illustrate this concept, a signal, $s(t)$, was generated using MATLAB. This signal was then corrupted with random noise on four different channels, $x_1(t)$, …, $x_4(t)$. The average of these four channels, $y(t)$, was taken to reduce the noise effects. The result of this simulation is shown in **Figure 2**. It can be seen that this method is effective in reducing, although not completely eliminating, the noise contamination in a signal.

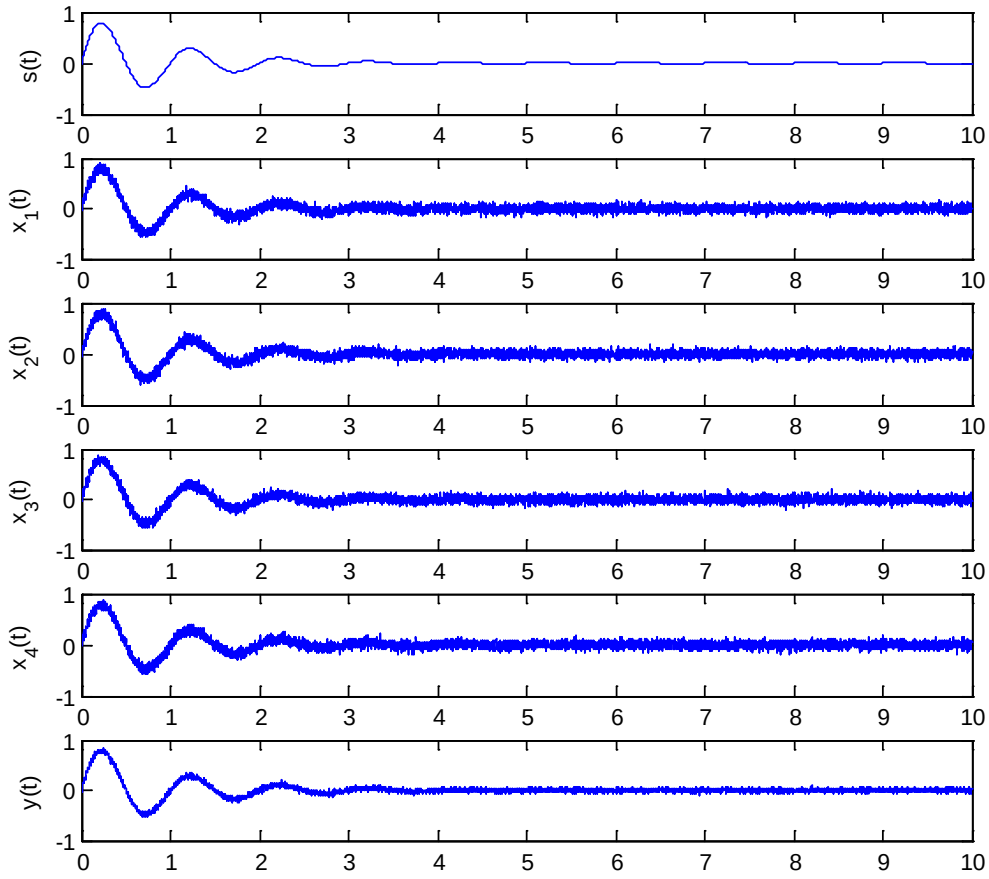

Figure 2. Noise Rejection Example

2.2.2 Beamforming

The concept of beamforming is applying different weights to each of the microphones in order to "shape" and "steer" the directivity pattern of the array (McCowan, 2001). Effectively this will aim the array to measure at a desired location. Beamforming techniques can be classified as either fixed (data-independent), or adaptive (data-dependent) (McCowan, 2001). Fixed beamformers have coefficients that remain fixed

during operation, and therefore are simpler than adaptive techniques. Delay-sum beamforming is the simplest example of a fixed beamformer. In this technique, each of the individual microphone channels, $x_i(t)$, is delayed in time by a particular time delay, τ_i, which is determined from the geometry of the array and the desired direction of measurement. Here $i = 1, 2, \ldots, n$, where n is the number of microphones in the array. These delayed time domain signals are then summed to obtain a single array output (McCowan, 2001). A diagram to illustrate this process is shown in **Figure 3**.

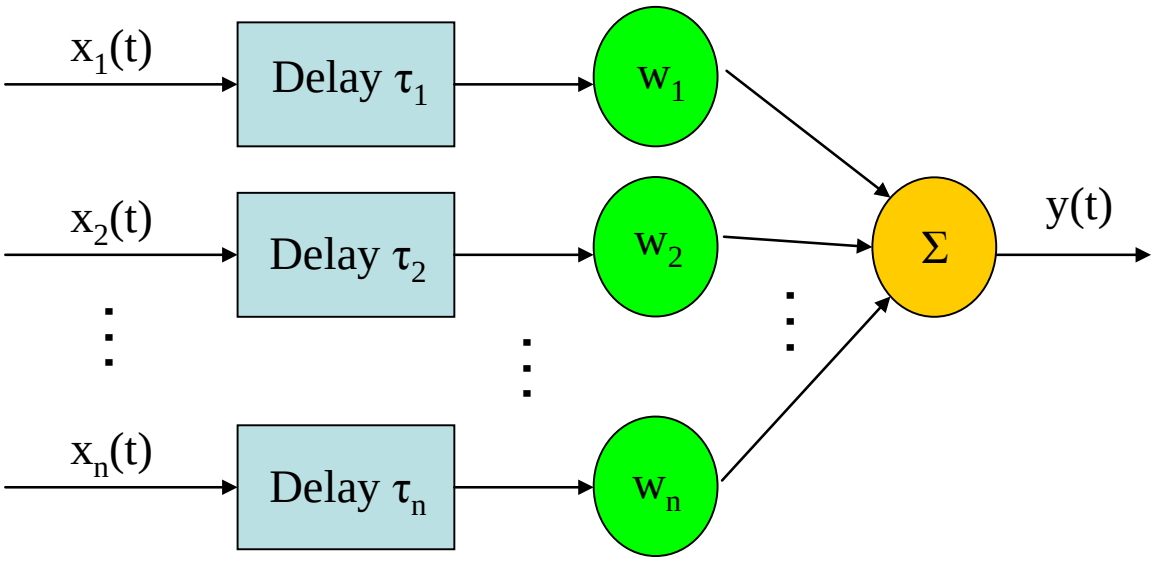

Figure 3. Delay-Sum Beamformer Diagram

Some other examples of fixed beamformers are sub-array delay-sum, superdirectivity, near-field superdirectivity, and filter-and-sum (McCowan, 2001, Johnson and Dudgeon, 1993).

Adaptive beamformers have signal processing coefficients that change over time to achieve or maintain a desired control parameter (Eargle, 2004). Some examples of adaptive beamforming methods are generalized sidelobe canceller, adaptive microphone-

array system for noise reduction (AMNOR), post-filtering, and worst-case performance optimization (McCowan, 2001, Vorobyov, Gershman, and Luo, 2003).

2.2.3 Tracking

Array processing techniques can be used to track the positions of certain objects. Tracking is useful in various applications, including military defense, maintaining safe distances between objects, and sound source localization. These types of systems can track an object as long as it is emitting or reflecting some sort of propagating signal. For example, a ship at sea will emit and reflect underwater sound waves. This information can be measured and interpreted by an array to determine signal characteristics such as direction of propagation and frequency content. If multiple arrays are used to measure this data, the object's position and velocity can be estimated (Johnson and Dudgeon, 1993). In the case of acoustics, a single microphone array can be used to find the angle of incidence of an acoustic event. If two or more microphone arrays at different locations capture the same acoustic event, the position of the event can be estimated. This is done by taking the intersection of each of the angle of incidence lines, as shown in **Figure 4**.

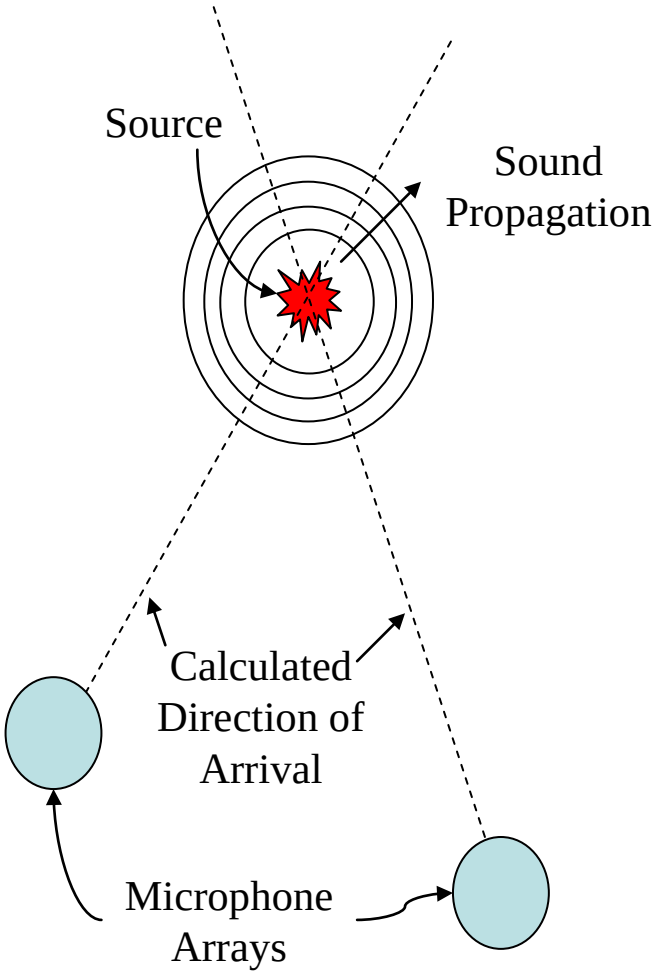

Figure 4. Finding Position with Multiple Microphone Arrays

2.3 CORRELATION FUNCTIONS

One of the advantages of recording multiple simultaneous channels of data, for example with a microphone array, is the ability to analyze the correlation between the different channels. Of particular use to this application are the autocorrelation function $R_{xx}(\tau)$ and the cross-correlation function $R_{xy}(\tau)$, of two signals $x(t)$ and $y(t)$, given by

(3) $$R_{xx}(\tau) = \frac{1}{T}\int_0^T x(t)x(t+\tau)dt$$

(4) $$R_{xy}(\tau) = \frac{1}{T}\int_0^T x(t)y(t+\tau)dt$$

Note that the auto-correlation function is just the cross-correlation function of a given signal with itself. That is, substituting *x(t)* for *y(t)* in equation (4) will yield equation (3). Note also that $R_{xy}(\tau) = R_{yx}(-\tau)$. It can be shown that the auto-correlation function is always an even function. The auto-correlation function represents time properties in the data that are separated by fixed time delays. For example, consider the auto-correlation function for a signal $x(t) = \sin(t)$ of length *T*, where *n* is an integer. For this example, equation (3) becomes

(5) $$R_{xx}(\tau) = \frac{1}{T}\int_0^T \sin(t)\sin(t+\tau)dt\,.$$

Using the trigonometric identity,

(6) $$\sin(u)\sin(v) = \frac{1}{2}\left[\cos(u-v) - \cos(u+v)\right],$$

and performing the integration,

(7) $$R_{xx}(\tau) = \frac{1}{2}\cos(\tau) + \frac{1}{4T}\left[\sin(\tau) - \sin(\tau + 2T)\right].$$

If the length of the signal T is a multiple of the period of the signal, *i.e.*, $T = 2\pi n$, where n is an integer, then the equation reduces to

(8) $$R_{xx}(\tau) = \frac{1}{2}\cos(\tau).$$

Therefore, for a sine wave, the auto-correlation function will have peaks at any time shift of $2\pi n$, where n is an integer. This corresponds to the signal aligning with itself due to its natural periodicity. An illustration of this result is given in **Figure 5**.

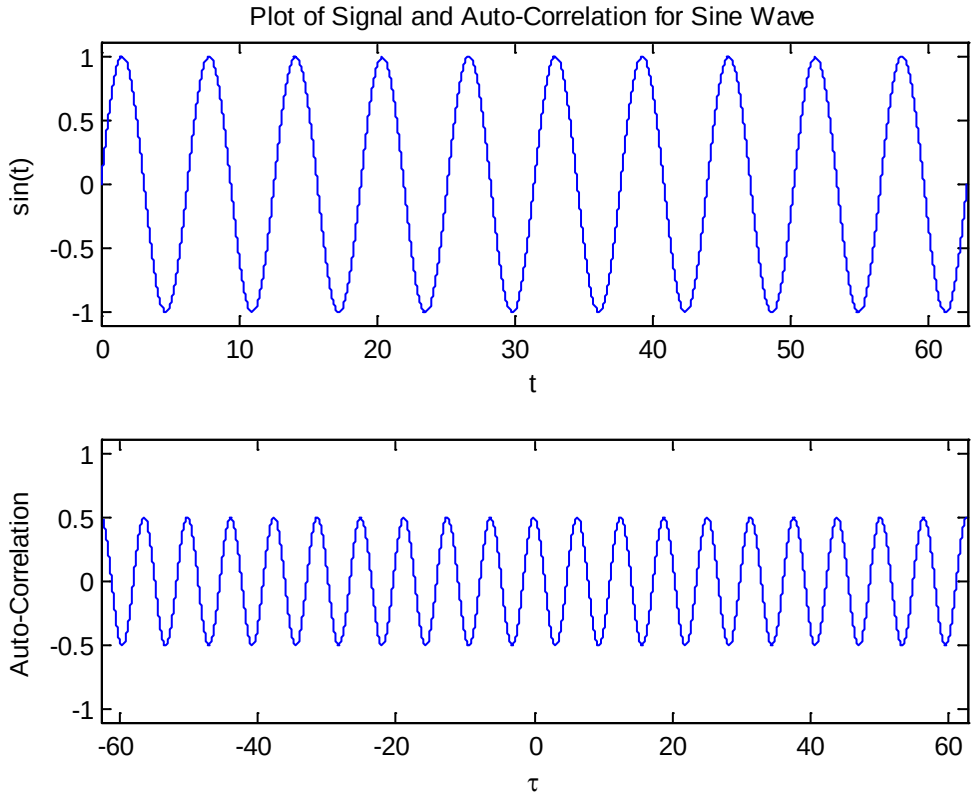

Figure 5. Plot of Signal and Auto-Correlation for Sine Wave

The auto-correlation of non-periodic signals such as random noise or impulse noise is much different than that of periodic signals. Consider, for example, data sampled from a standard normal distribution, *i.e.*, random noise with mean of zero and standard deviation equal to one. The auto-correlation for this signal would only align with itself when there is no time shift ($\tau = 0$). When τ is nonzero, the randomness of the signal will force the value of $R_{xx}(\tau)$ to approximately zero. This result is displayed in **Figure 6**.

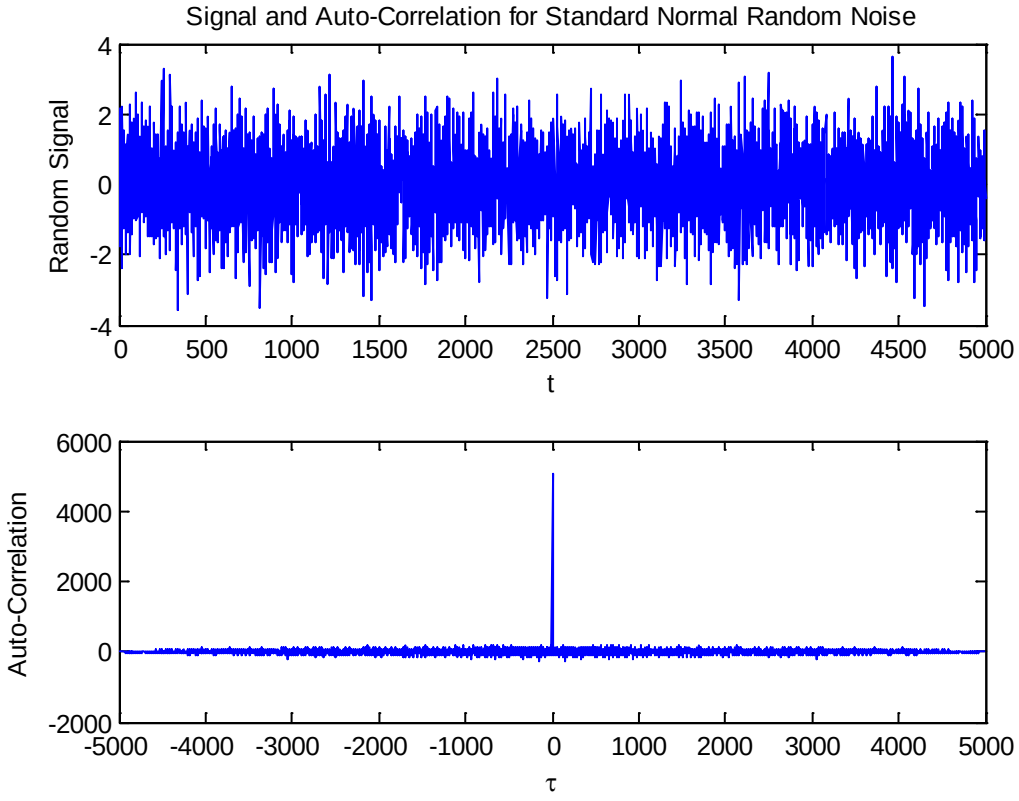

Figure 6. Plot of Signal and Auto-Correlation for Standard Normal Random Noise

Another interesting type of signal to consider is an impulsive signal. Of particular interest for this project is a military impulse signal, which will typically have one large peak followed by some oscillation and subsequent smaller peaks. The auto-correlation for a signal of this type will have some peaks around $\tau = 0$ due to the alignment of the absolute maximum with the subsequent local maxima and minima. To help illustrate this, **Figure 7** shows an example impulse wave from actual collected data. This waveform was created from an M-15 anti-tank mine detonated at a distance of approximately 150 meters from the recording location.

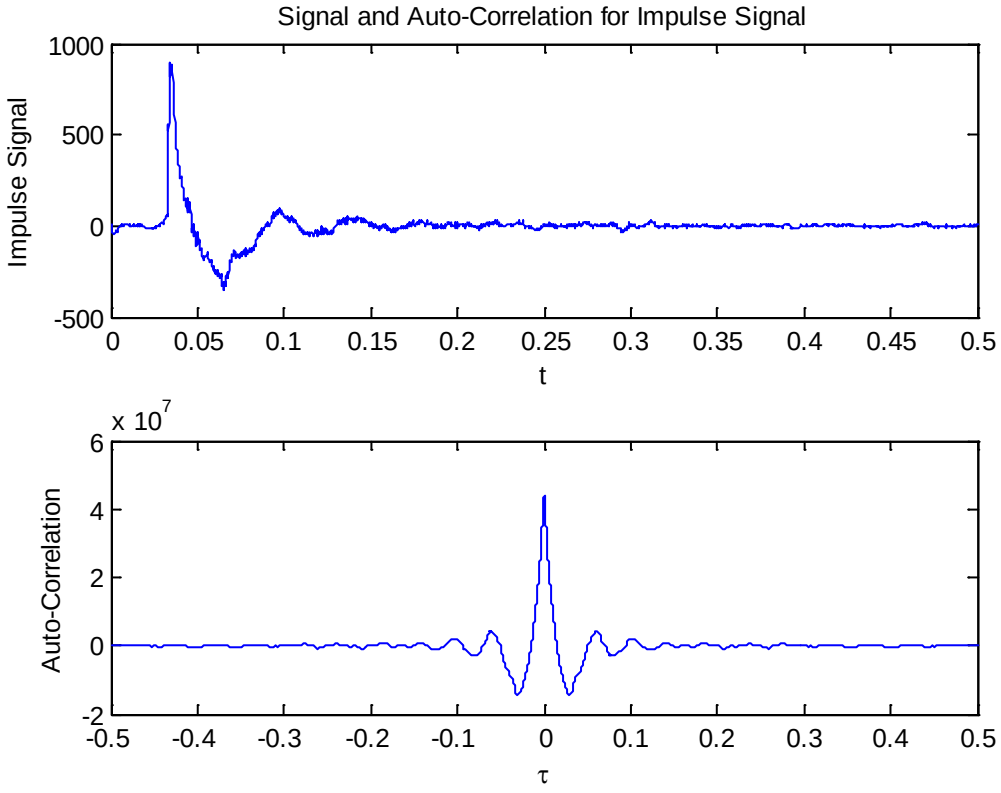

Figure 7. Plot of Signal and Auto-Correlation for a Military Impulse Signal

The cross-correlation function between two signals represents their similarity based on a time shift, τ. This can be useful for various applications, including detection of time delays, noise source identification, and radar and sonar applications (Norton, 2003). In general, unlike the auto-correlation function, the cross-correlation function will not be an even function. In fact, one useful feature of this function is measuring the value of τ where the function is at its absolute maximum. This value represents a time delay between the two signals. To help illustrate this concept, using the previous impulse signal from **Figure 7**, another signal was created by delaying the original signal by 0.1 seconds. **Figure 8** shows the original and delayed signals together, as well as their cross-

correlation function. The absolute maximum of the cross-correlation function occurs at a value τ = -0.1 *s*, which agrees with the time delay between the signals. Here the minus sign indicates that there is a lag in time as opposed to a positive sign which would represent a lead in time. Notice that the cross-correlation function for this example is the same as the auto-correlation function in **Figure 7**, except for the shift in time.

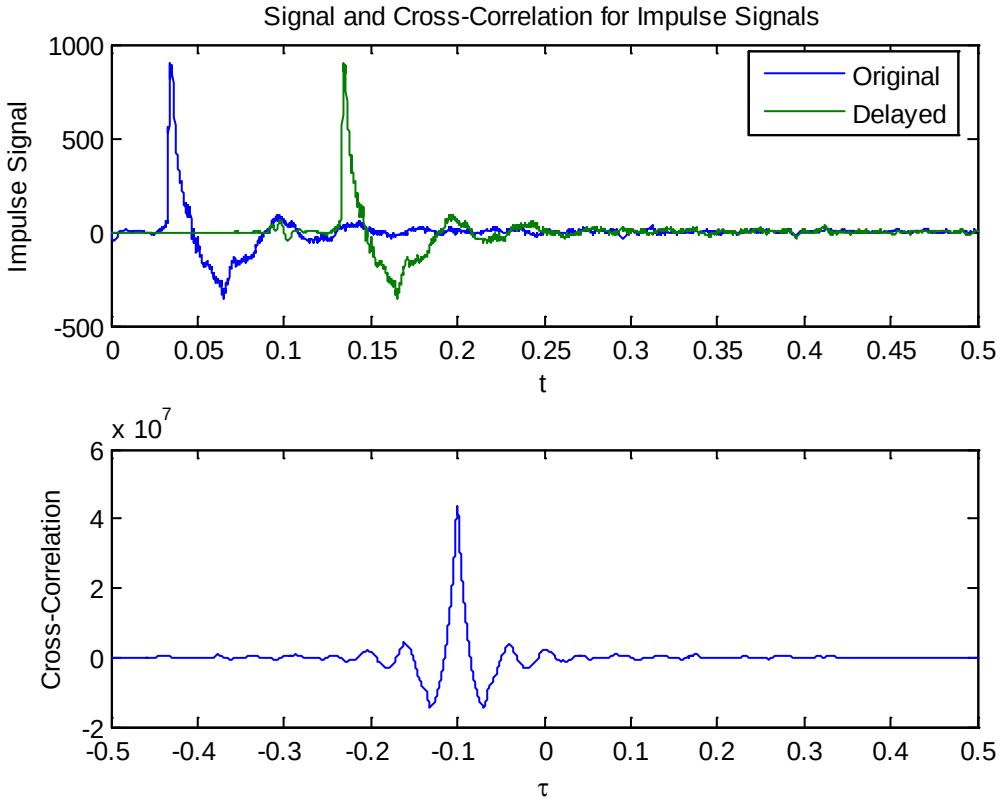

Figure 8. Plot of Original and Delayed Signals with Cross-Correlations

The calculation of the auto-correlation and cross-correlation functions can be visualized by a shifting in time of one signal with respect to one another, while

multiplying their corresponding values. This process is similar to that of a convolution integral of two functions *x(t)* and *h(t)*, given by (Rao, 2004)

$$y(\tau) = \int_0^{\tau} x(t)h(\tau - t)dt \tag{9}$$

The primary difference between a convolution integral and a cross-correlation is the negative sign in the convolution integral, which in effect is a reflection of one of the signals about $t = 0$ before time shifting.

When dealing with discrete or digital data, the auto-correlation and cross-correlation functions cannot be calculated exactly. In this case, equations (3) and (4) can be approximated as (Bendat and Piersol, 1986)

$$\hat{R}_{xx}(r\Delta t) = \frac{1}{N-r}\sum_{n=1}^{N-r} x_n x_{n+r} \tag{10}$$

$$\hat{R}_{xy}(r\Delta t) = \frac{1}{N-r}\sum_{n=1}^{N-r} x_n y_{n+r} \tag{11}$$

where $r = 0, 1, 2, \ldots, m$, with $m < N$. Here, r represents the lag number, and Δt is the time step, which is the inverse of the sampling frequency in Hz. The ^ symbol above the R denotes an approximation. Another quantity of interest is the cross-correlation coefficient function $\rho_{xy}(\tau)$, given by

(12) $$\rho_{xy}(\tau) = \frac{R_{xy}(\tau)}{\sqrt{R_{xx}(0)R_{yy}(0)}},$$

which can be approximated by

(13) $$\hat{\rho}_{xy}(r\Delta t) = \frac{\hat{R}_{xy}(r\Delta t)}{\sqrt{\hat{R}_{xx}(0)\hat{R}_{yy}(0)}}.$$

This quantity always satisfies $-1 \leq \rho_{xy}(\tau) \leq 1$ (Bendat and Piersol, 1986, 1993). Since the cross-correlation coefficient is a normalized quantity, it is ideal for use in comparing different waveforms.

3.0 DATA COLLECTION

At the onset of development of this noise classifier, it was necessary to obtain a library of measurements containing military impulse noise and potential false positive noise sources for the classifier. A library of single channel data was collected by Bucci and Vipperman using the following methods. The necessary data collection requirements were not met by any commercially available data collection system, so a composite system was constructed from a few different components. Most of the energy of the measured noise sources falls between 0 and 100 Hz, so a B&K 4193 infrasonic microphone was selected, due to its bandwidth of 70 mHz to 20 kHz. This microphone was connected to a Larson Davis (LD) 824 Sound Level Meter (SLM). The AC output of the LD 824 SLM was then connected to a National Instruments (NI) BNC-2110 input/output board and then into a NI DAQCard-6036E data acquisition card. The data acquisition card was installed in a Dell Latitude laptop with a Pentium IV processor. A Virtual Instrument (VI) was created to capture waveform data. **Figure 9** shows an example screen of the VI.

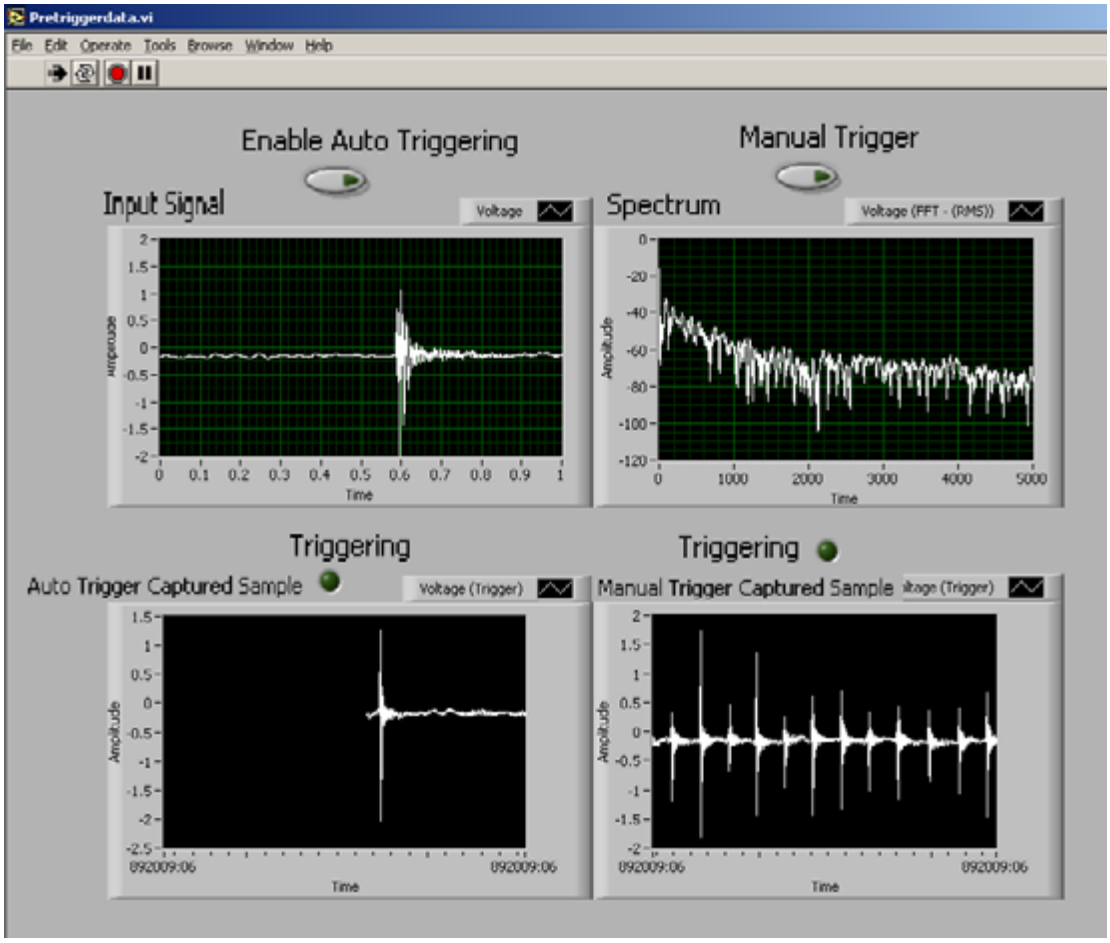

Figure 9. Example Screen of Virtual Instrument

The VI operates using an "automatic triggering/pre-triggering mode." This mode allows for the automatic recording of all events that exceed a given threshold level, corresponding to a particular peak value, L_{pk}, defined by (Norton, 2003)

$$L_{pk} = 20 \cdot \log_{10}\left(\frac{p_{pk}}{p_{ref}}\right) \tag{14}$$

where p_{pk} is the peak sound pressure in Pascals (Pa), p_{ref} is 20 μPa, and L_{pk} is the peak level in decibels (dB). The pre-trigger will record a certain amount of data before the event trigger, which allows the recording of the entire event. Although the threshold can be set to record at or above any desired L_{pk} level, it was typically adjusted to just above ambient noise levels in order to record as much useful data as possible. In the data collection, a 0.5 second pre-trigger was coupled with an additional 2 seconds of recorded data for each record. A manual triggering mode was also used to record longer or continuous events such as wind, aircraft noise, traffic, and engine noise. The "automatic mode" was also able to capture multiple successive trigger events, as it triggered nearly continuously. Although the majority of the military impulse noise energy would be between 0 and 100 Hz, some noise sources, such as aircraft, have significant energy up to 3 kHz. To ensure the accurate capture of all noise sources, data were sampled at 10 kHz (Bucci and Vipperman, 2006, 2007, Bucci, 2007). Data were collected at seven different military bases throughout the United States. These bases are numbered from one to seven. Single channel data were collected at Bases 1 – 6. **Figure 10** shows a picture of the instrumentation setup for single channel data collection.

Figure 10. Single Channel Data Collection Field Setup

To further the work by Bucci and Vipperman, additional data were gathered using a microphone array to collect four simultaneous channels of sound pressure data. The geometry of this array consists of three in-plane microphones equally spaced in a circle

with radius of one meter. A fourth microphone is placed one meter from the center of that circle perpendicular to the plane. During data collection, the in-plane microphones are positioned parallel to the ground, with the perpendicular microphone pointing upwards (Rhudy *et al.*, 2009). Each of the four microphones used was a PCB Electronics 377B02 ½" pre-polarized free field microphone connected to a PCB Electronics 426E01 preamplifier. As in previous work by Bucci and Vipperman, the microphones were connected through a National Instruments BNC-2110 input/output board into a National Instruments DAQCard-6036E data acquisition card which was installed in a Dell Latitude laptop with a Pentium IV processor. **Figure 11** shows a picture of the microphone array field setup.

Figure 11. Microphone Array Data Collection Field Setup

Data were collected over two separate trips to Base 7. Measurements were taken at three different locations on the base. The data totals four channels of recording of 86 Bradley (25 mm) and 146 wind waveforms. The Bradley waveforms had L_{pk} values ranging from 71 dB to 131 dB, and the wind waveform L_{pk} values were in the range of 72 dB to 102 dB (Rhudy *et al.*, 2009).

In order to develop a robust classifier, it was desired to obtain data under various conditions. Different locations for recording sessions were selected to provide various terrain types and distances. Single channel data collection was conducted at six different military bases. Data were taken at locations ranging approximately from 150 m to 6 km from the military impulse noise source. The terrain for recording sessions included flat open fields, mountainous dense forest, over soft sandy soil, on top of rocky terrain, tops of ridge-lines, valley bottoms, and beside large bodies of water (Bucci and Vipperman, 2006, 2007). A reasonable range of weather conditions was obtained, with temperatures ranging from 52 to 100 °F and relative humidity varying from 44 to 100 % and raining (WxUSA U.S. Weather Reports). In addition to different environmental conditions, different types of waveforms were gathered. These waveforms included various types of military impulse noise of interest, as well as non-impulsive noise, which consisted of vehicle and wind noise. The types of waveforms collected are summarized in **Table 1**.

Table 1. Summary of Collected Waveforms

Types of Noise Source	Number of Recordings
120mm M1 Tank Main Gun	234
25mm Bradley Chain Gun	137
155mm Howitzer Firing	244
155mm High Explosive Projectile	36
70mm Hydra Rocket	2
81mm Mortar Impact	126
M67 Hand Grenade	5
Bangalore Torpedo (string of 3)	4
M19 Landmines	7
M15 Landmines	8
2 x M19 Landmines	1
2 x M15 Landmines	3
Small Arms	41
F-16 fly over	204
C-5 Cargo Plane	1
A-10 Fly over	6
Helicopter	105
Other Vehicle Noise	104
Wind noise	1402
Total	**2670**

4.0 SINGLE CHANNEL DATA ANALYSIS

Early work in the development of this military impulse classifier by Bucci and Vipperman utilized only a single channel of data. After obtaining a library of waveforms, various analyses were conducted on the data in order to determine different characteristics of the data set. Different classification methods were also explored to determine which classifier would be most effective (Bucci and Vipperman, 2006, 2007, 2008, Bucci, 2007).

4.1 VERIFICATION OF INVERSE SQUARE LAW FOR ACOUSTICS

The data collection trip to Base 6 provided a new level of control in the measurement conditions. The impulse waves were generated by detonating either single or double M-15 or M-19 anti-tank mines. Data were collected in direct collaboration with the detonation team. The detonation site remained the same while the data collection team moved to different measurement locations, which varied in distance from the detonation site. 19 usable waveforms were collected. The four different test cases are summarized in **Table 2**. Each test case included a detonation charge in addition to the high explosive content in the mines.

Table 2. Summary of Controlled Demolition Charges Recorded at Base 6

Test Case	Quantity	Description	High Explosive Content (lb)
A	7	M-19 anti-tank mine	21
B	1	M-19 anti-tank mine (x2)	42
C	8	M-15 anti-tank mine	22.5
D	3	M-15 anti-tank mine (x2)	45

In addition to the 19 collected waveforms, data was obtained from the 3rd-generation noise monitors around the detonation site. Only some of the blasts were measured by the monitors, due to their higher triggering threshold. However, 12 of the blasts were measured by at least one monitor. This data was used to compare with the collected data. **Figure 12** shows L_{pk} versus distance plots corresponding to the four individual test cases presented in **Table 2**. **Figure 13** represents this same data, but combined all into one graph.

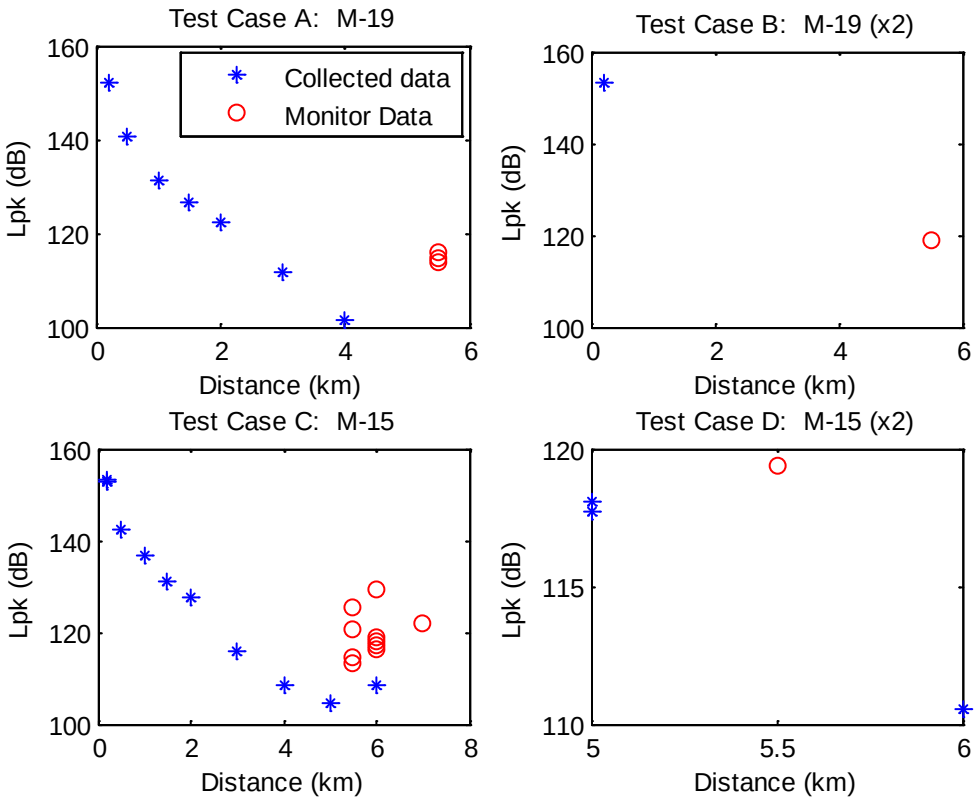

Figure 12. Comparison of Collected Data and Monitor Data by Test Case

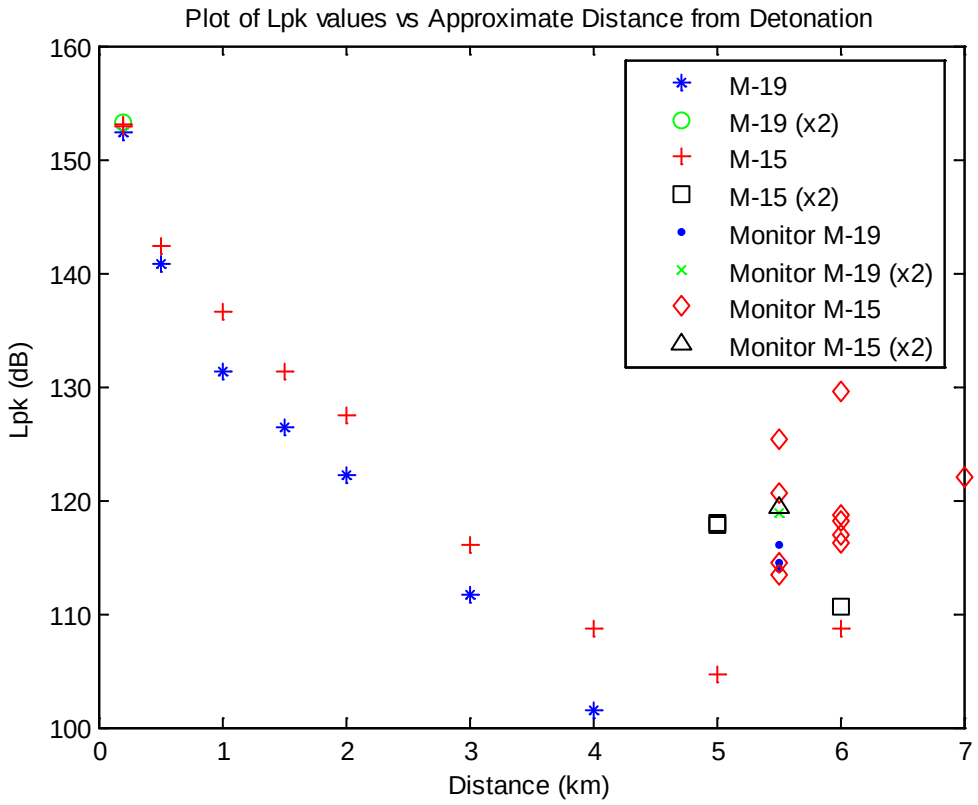

Figure 13. Comparsion of Collected Data and Monitor Data for all Test Cases

Since data were collected at multiple distances for test cases A and C, the inverse square law for acoustics can be used to assess the measurement integrity. The inverse square law states that sound pressure will be inversely proportional to the distance from the source (Crocker, 1998). In order to check if the data followed this trend, the peak sound pressure was plotted with respect to the inverse of distance. Then a least-squares linear curve was fitted to the data. This procedure was done for both test case A, as seen in **Figure 14**, and test case C, as seen in **Figure 15**. The linear curve fits yielded R^2 values close to one, indicating a good linear fit. **Figure 16** and **Figure 17** show the original data plotted with the inverse square law curve fit for test cases A and C

respectively. From here it can be seen that the collected data for test cases A and C both follow the inverse square law very well.

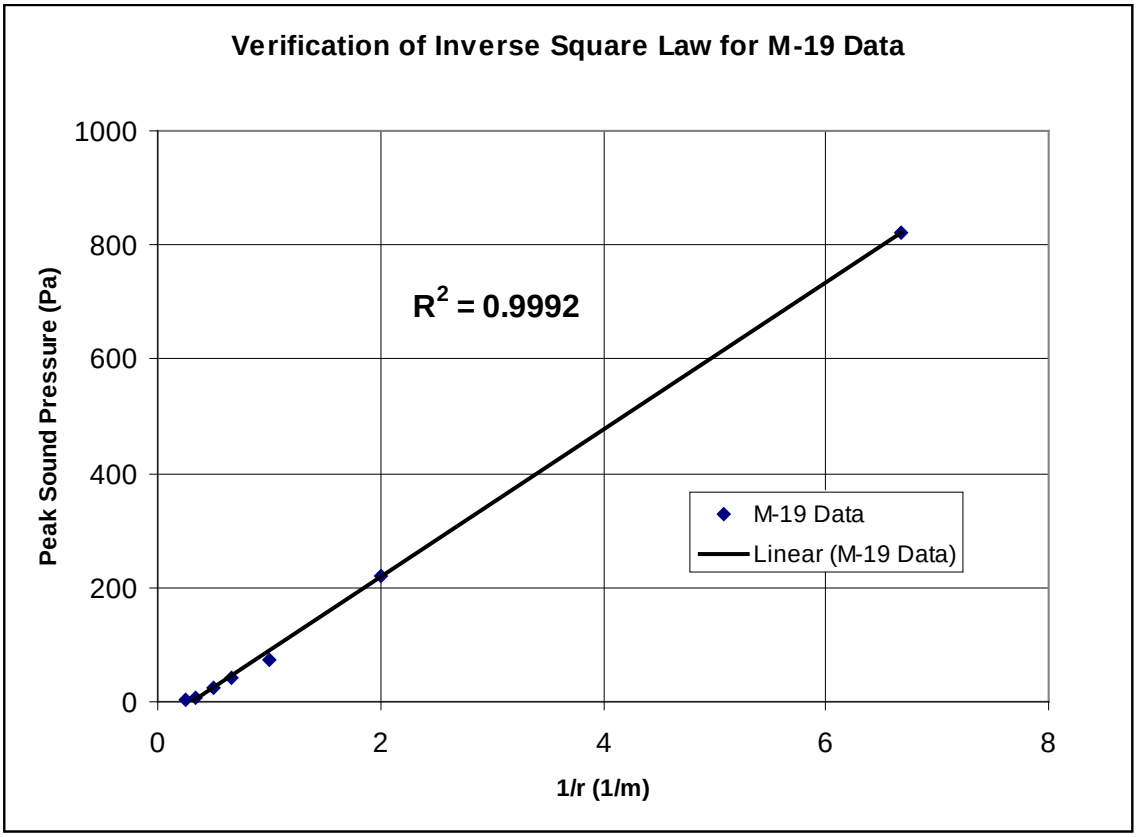

Figure 14. Least Squares Linear Curve Fit for Test Case A

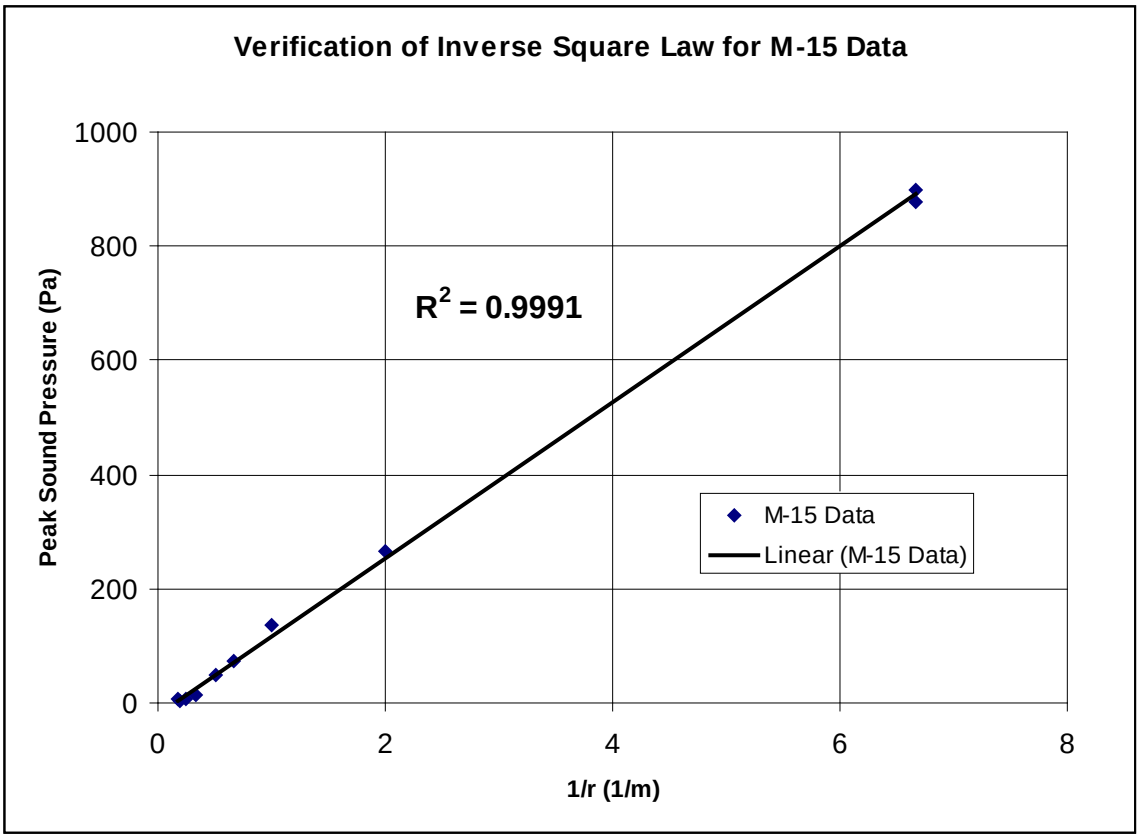

Figure 15. Least Squares Linear Curve Fit for Test Case C

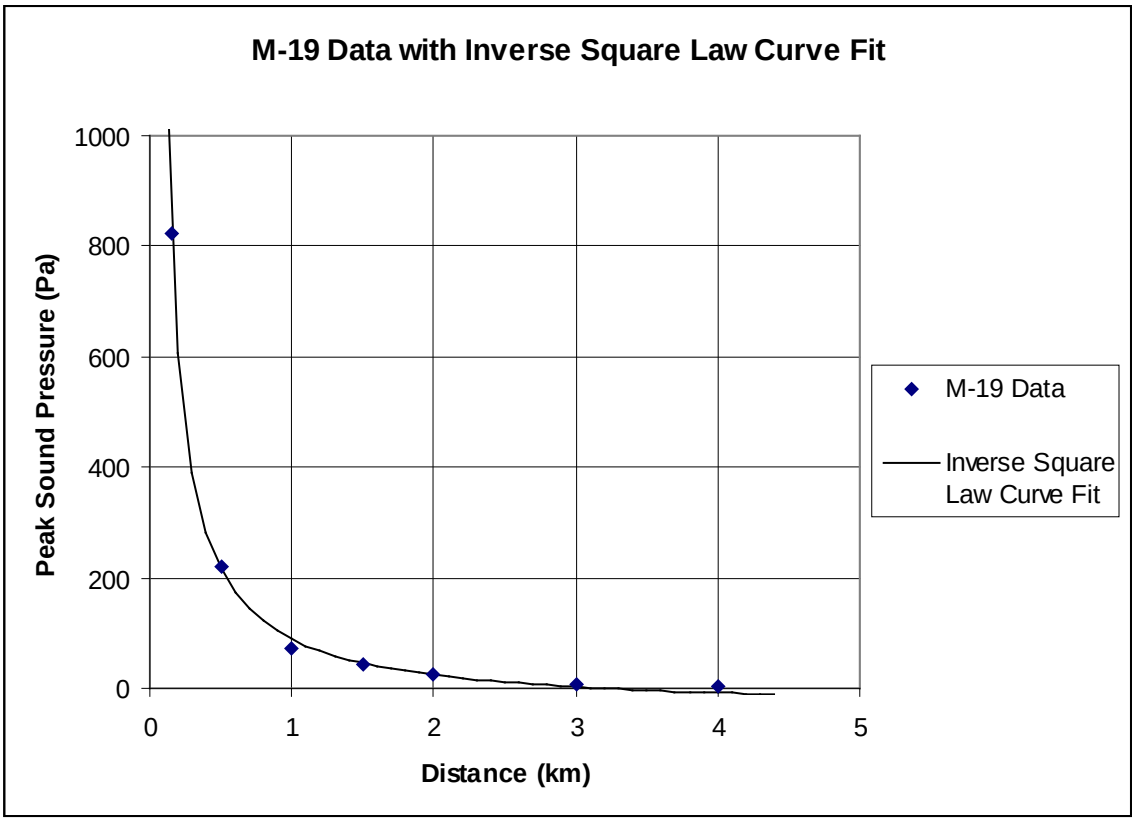

Figure 16. Test Case A with Inverse Square Law Curve Fit

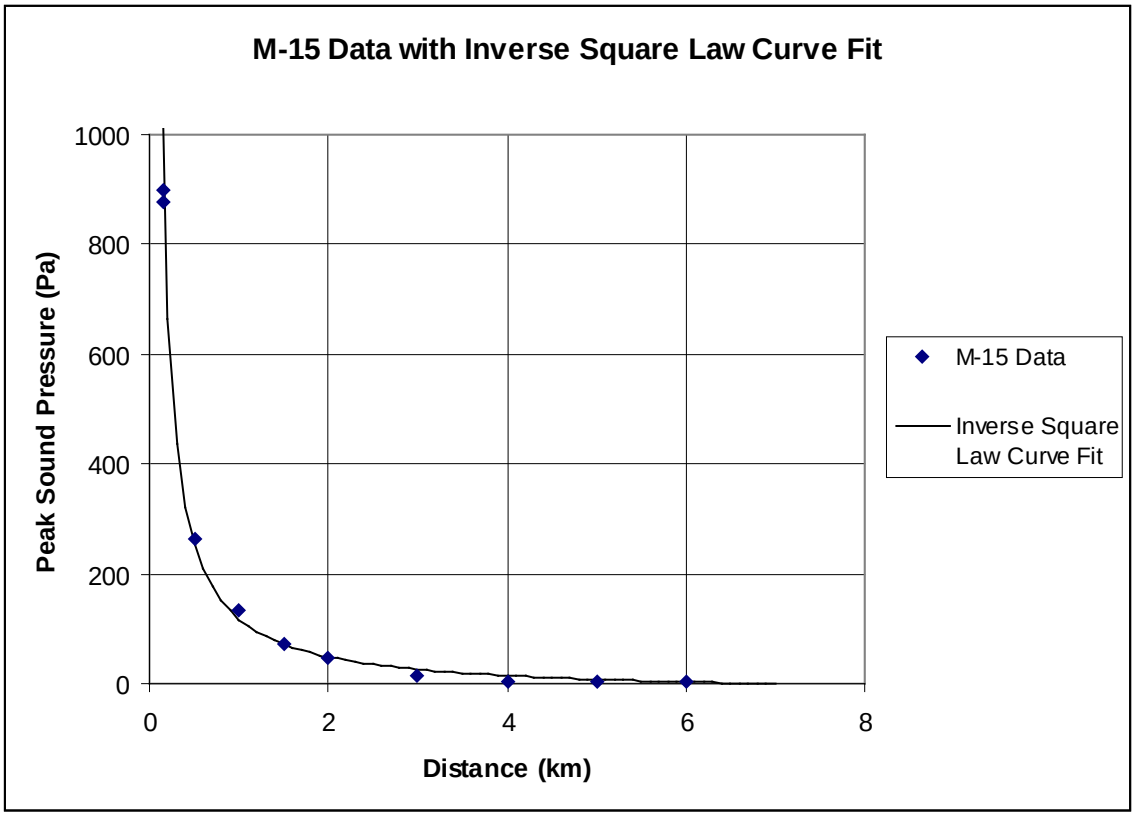

Figure 17. Test Case C with Inverse Square Law Curve Fit

4.2 FALSE POSITIVE WIND TRIGGERING ANALYSIS

Base 3 is known to experience high wind speeds, therefore data collection at this location provided significant insight into the issue of false positive impulse detection due to wind noise. There are multiple active 3rd-generation noise monitors placed at various locations throughout the base. During the collection trip, impulse noise was unable to be heard at the noise monitors around the perimeter of the base, so data were collected at locations closer to the training areas. The types of recorded military impulse waveforms were 120 mm tank rounds and 25 mm Bradley rounds. During the period of recording, wind gusts

of up to 49 mph were present. Two interior noise monitors measured several impulse events. Data were collected at the same location as one of these monitors, which was near the tank firing range. **Figure 18** shows the results of these measurements. Wind speed and wind gust data was obtained and plotted with respect to the right ordinate axis (WxUSA U.S. Weather Reports). L_{pk} values from both the monitors and the collected data are plotted with respect to the left ordinate axis. In general, it can be seen that more events were detected by the data collection setup than were detected by the existing noise monitors. This most likely occurred due to the higher noise threshold imposed on the noise monitoring system. The L_{pk} discrepancies between the University of Pittsburgh collected data and the on-site monitors was also observed at other military bases and could not be explained.

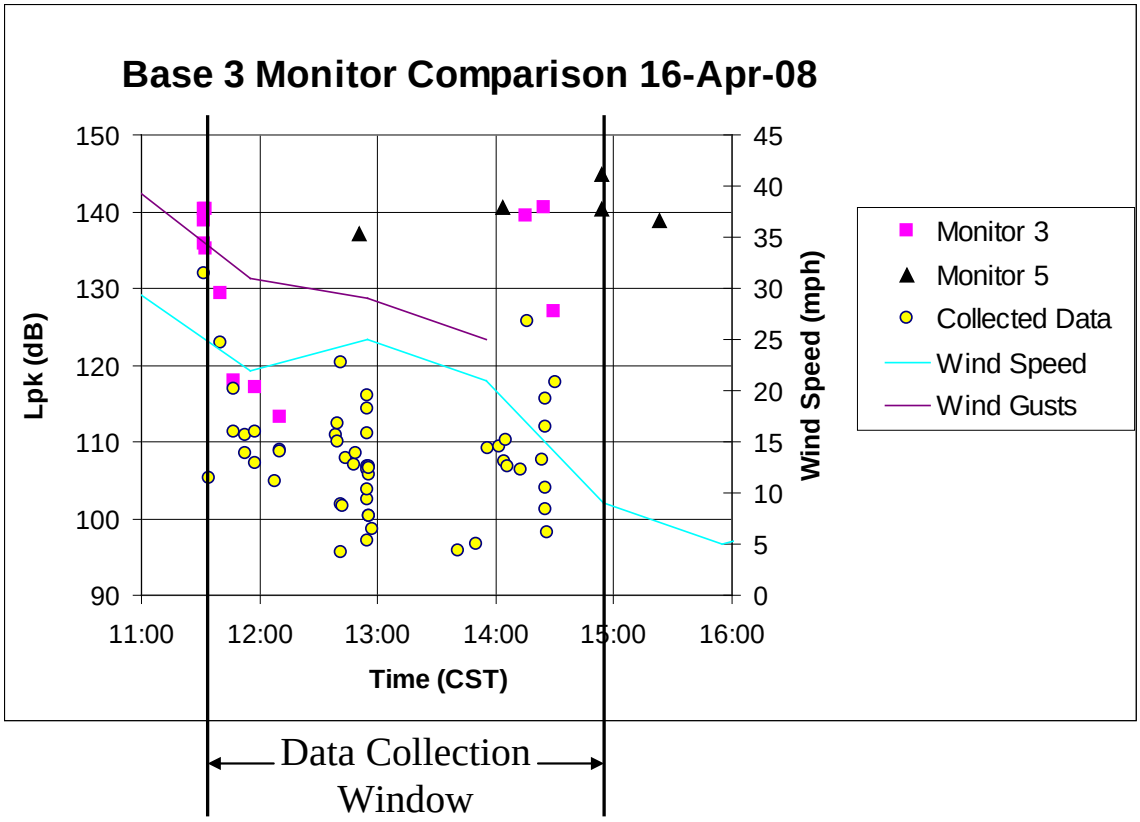

Figure 18. Comparison of Monitor and Collected Impulse Data from 11:00-16:00 CST on 16-Apr-08

The following two figures help illustrate the issue of false positive impulse triggers due to wind noise at Base 3 in the monitor units. **Figure 19** shows the same representation of data as **Figure 18**, except the time scale is extended to the entire day, and the data were extended to include all of the functional monitors. Data collection was only conducted during the time frame of 11:30 – 14:30 CST, but training presumably occurred before and after this window. High winds were experienced from 0:00 to 12:00 CST, during which time all of the monitors detected events. Monitor 5, which is located near the impact area, detected very high level events. These are most likely impulse noise, due to the location of the sensor. Since impulse noise was not heard at the

monitors on the perimeter of the base, it is likely that many of the triggers between 0:00 and 12:00 CST at those monitors are due to wind, as corroborated by the high wind levels shown in **Figure 19**.

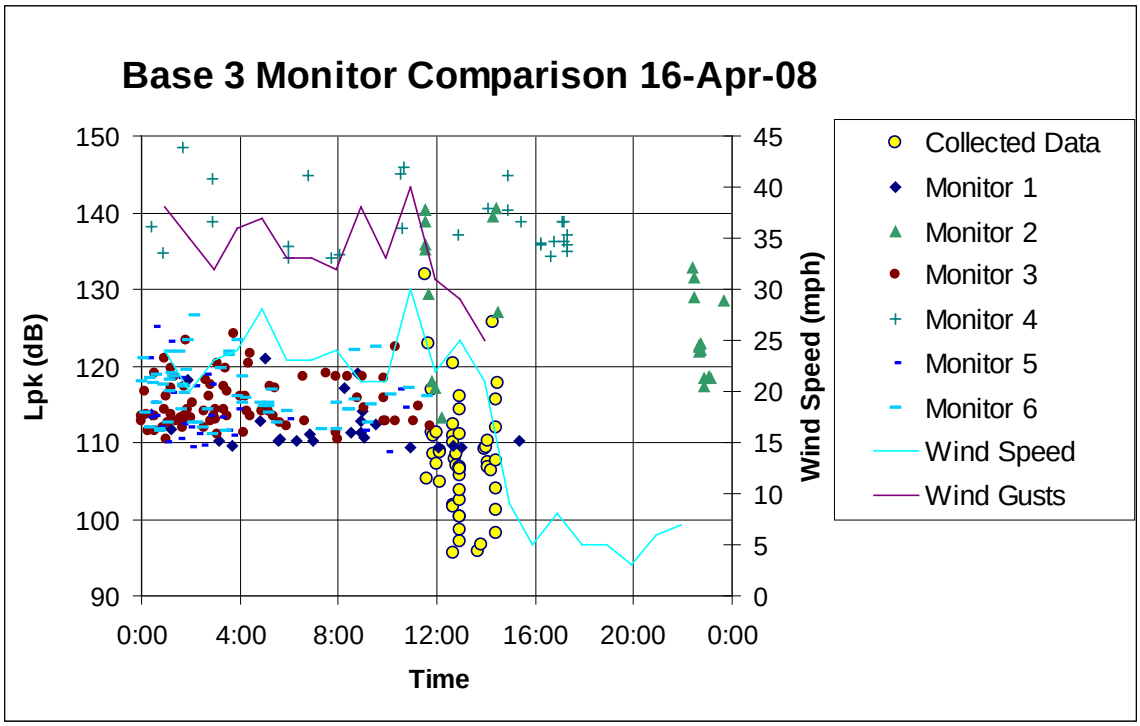

Figure 19. Comparison of Monitor and Collected Data from 0:00-24:00 CST on 16-Apr-08

Figure 20 shows all of the noise monitor data for Sunday, 21-Apr-08. The training logs for that day reported only small arms fire. The two largest ordnance weapons fired were a .50 cal and 12-gauge shotgun. Due to the low acoustic levels that these weapons produce, they would not be sensed by monitors 2, 4, 5, and 7, which are located on the perimeter of the base. However, the monitor logs show multiple triggers in these locations. Monitor 5 measured notably high sound levels, which could be explained by the fact that it is located in an open field, which provides no protection from

the wind. It can be seen in **Figure 20** that the wind speeds have significant correspondence with the wind triggering of the monitors.

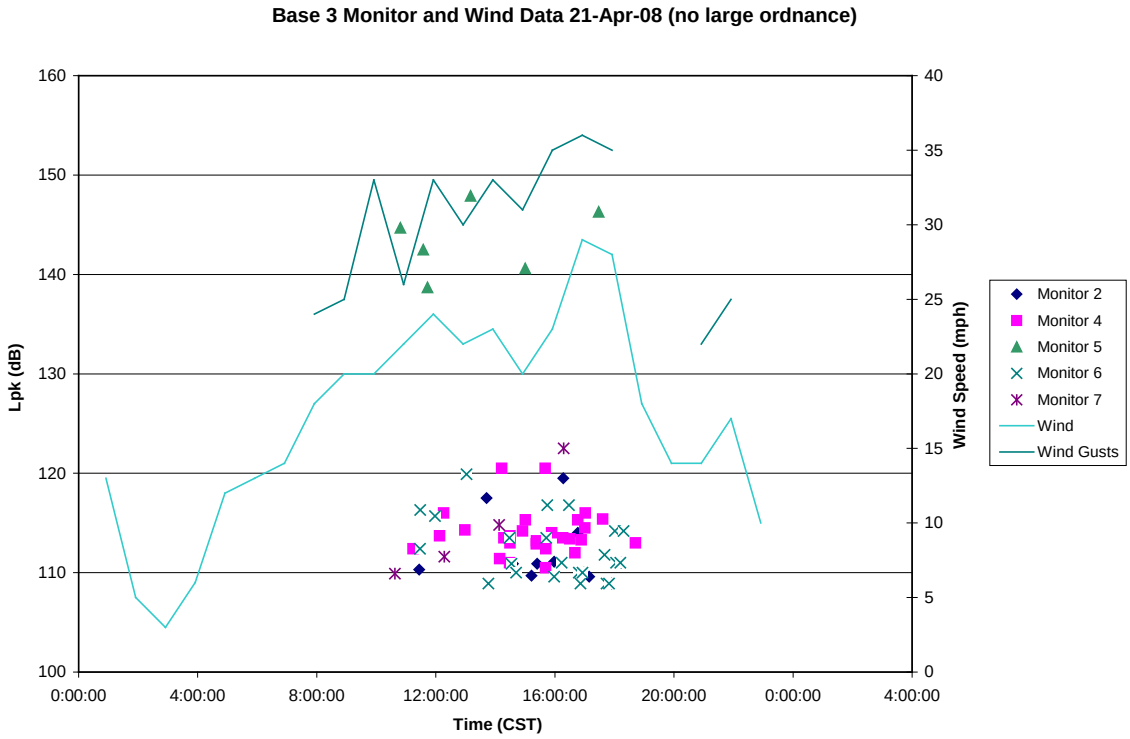

Figure 20. Comparison of Wind Speeds and Monitor Triggers on 21-Apr-08

4.3 ARTIFICIAL NEURAL NETWORK (ANN) CLASSIFIER

In order to determine whether a given input waveform is impulsive or non-impulsive, a multi-layer perceptron (MLP) ANN (Haykin, 1999) was constructed. The structure of this ANN was optimized heuristically, which is common practice. The inputs for the ANN are the four scalar metrics defined in section 4.3.1 (kurtosis, crest factor, spectral slope, and weighted square error). The ANN contains three hidden layers, each with four

nodes, and one output node. The Levenberg-Marquardt algorithm was used to train the ANN (Chong, 2001). The output of the network is a single scalar number between zero and one. Outputs closer to zero indicate non-impulse noise and outputs closer to one indicate impulse noise. Typically, 0.5 was used as threshold level to distinguish between non-impulse and impulse noise. However, this threshold can be shifted one way or the other to adjust the ratio of false positives (FPs) and false negatives (FNs) (Bucci and Vipperman, 2006, 2007, Bucci, 2007).

4.3.1 Scalar Metrics

Previous work by Bucci and Vipperman utilized four scalar metrics that are effective in classifying military noise. Two of these metrics, kurtosis and crest factor (CF), are widely used in acoustics and vibration analysis to characterize waveforms. Both of these metrics tend to be good indicators of impulsiveness of a signal (Norton, 2003). The other two metrics, spectral slope (m) and weighted-square error (WSE), were developed from observations made from the power spectral density (PSD) functions calculated from the data. Crest factor (CF) is calculated by

$$\mathrm{CF} = \frac{\left| p_{pk} \right|}{p_{rms}}, \tag{15}$$

where $|p_{pk}|$ is the maximum absolute instantaneous pressure amplitude, and p_{rms} is the RMS of the sound pressure over some period of time. Kurtosis is the fourth central statistical moment given by

(16) $$\text{Kurtosis} = \frac{E\left[(p-\mu)^4\right]}{\sigma^4} = \frac{1}{\sigma^4 T}\int_0^T (x-\mu)^4\, dt,$$

where μ is the mean acoustic pressure (typically zero), σ is the standard deviation of the signal, and T is the time frame over which kurtosis is computed (Norton, 2003). The spectral slope (m), is computed by creating a least-squares fit to a line,

(17) $$\hat{y} = mx + b,$$

where $\hat{y}$ =$\log_{10}$(PSD) is the base-10 logarithm of the power spectral density and x = $\log_{10}(f)$ is the base-10 logarithm of frequency. The fit is conducted over the frequencies between 2.5 and 100 Hz. These frequencies are considered because most of the energy for impulse noise (Benson, 1996, Attias et al., 2004) and wind noise (Blevins, 2001) is in this low frequency range. Although the impulse and aircraft noise sources are poorly represented by a linear trend, the slope still provides useful information for differentiating from other types of noise. For the given spectral resolution (4,096 point FFT), there are 42 frequency bins from 2.5 to 100Hz. The weighted square error (WSE) is computed from the first 41 of these bins. To complement metric m, the "goodness" of the linear fit is assessed with the WSE, which is computed as

(18) $$WSE = \sum_{i=1}^{41}\left[y_i - \hat{y}_i\right]^2\left[f_{i+1} - f_i\right],$$

where y_i is based upon the $\log_{10}(\text{PSD}_i)$ of the i^{th} frequency bin, $\hat{y}_i$ is the estimate of y_i from the linear curve fit, and f_i is the base-10 logarithm of the i^{th} frequency. Squaring the quantity $\left[y_i - \hat{y}_i\right]$ forces WSE to be positive and also reflects the total magnitude of the error. The term $\left[f_{i+1} - f_i\right]$ adds greater weight to the error at the lower frequency bins, because the most effective classification features for distinguishing military impulse noise from non-impulse noise occur at the lower frequencies of interest. Another reason for this term is that other sources of environmental noise can contaminate the higher frequency bins of interest. The logarithmic PSD terms, y_i, are normalized between 0 and 1, which correspond to the minimum and maximum values of $\log_{10}(\text{PSD})$, respectively. This is done in order to scale the metrics with respect to signal energy. Thus, y_i is computed as

$$y_i = \frac{\log_{10}\left(PSD_i\right) - \min\left[\log_{10}\left(PSD\right)\right]}{\max\left[\log_{10}\left(PSD\right)\right] - \min\left[\log_{10}\left(PSD\right)\right]}. \tag{19}$$

It is important to note to the values of $\hat{y}_i$ are not normalized, but simply represent the curve fit of y_i (Bucci, 2007).

4.3.2 Determination of Dynamic Range

In order to determine the lower threshold of the dynamic range of the system, the ANN was trained using data after imposing various L_{pk} thresholds. This means that all data with L_{pk} values below the given threshold are eliminated from the data set before training and testing the network. The data set used for this analysis included all of the single channel data that were collected, which totaled 2263 waveforms. Two-thirds of the data

were used as training data, and the remaining one-third of the data was used as test data. In general, performance improved as expected when increasing the L_{pk} threshold. In fact, 100% accuracy is obtained when a threshold of 110 dB is applied to the data. As the threshold is increased, only waveforms with L_{pk} values exceeding that threshold are used to train and test the ANN. These higher level waveforms typically have a higher signal-to-noise ratio (SNR), which in general means they will be easier to classify, as they are less corrupted by undesired noise. Illustrations of the ANN performance with different imposed thresholds are seen in **Figure 21** through **Figure 25**. For these figures, it is desired to select a decision cutoff in order to minimize the number of errors (both false positives and false negatives). However, there is typically going to be a tradeoff when selecting this cutoff value, as moving the cutoff closer to one will typically cause a decrease in false positives, but an increase in false negatives. Similarly, if the cutoff is moved closer to zero, typically false negatives will decrease, but false positives will increase. *E.g.*, for **Figure 21**, there is no possible cutoff that can be selected in order to obtain zero errors. For the best possible overall accuracy, a cutoff should be selected between 0.80 and 0.85. Any cutoff selected within this region should lead to the same classification accuracy for this data set. As can be seen in **Figure 24** and **Figure 25**, perfect accuracy can be obtained if the cutoff is selected in a certain range, as indicated on the graphs. **Table 3** summarizes the results from all of these figures.

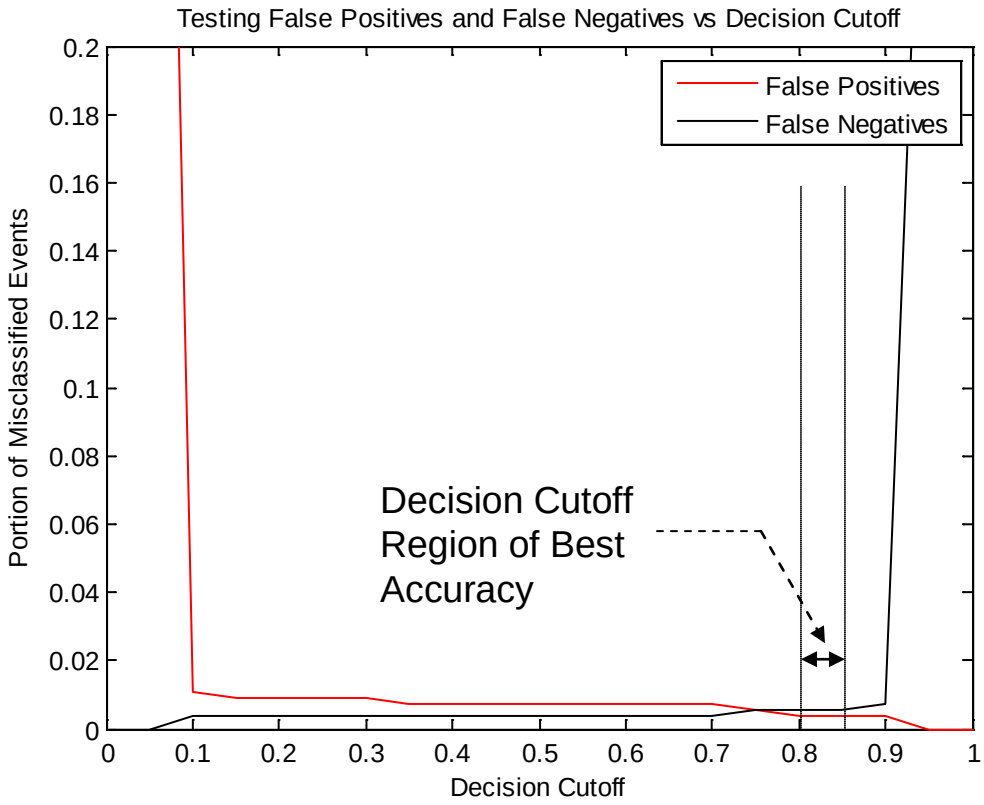

Figure 21. Accuracy of ANN Classifier with No Specified Threshold

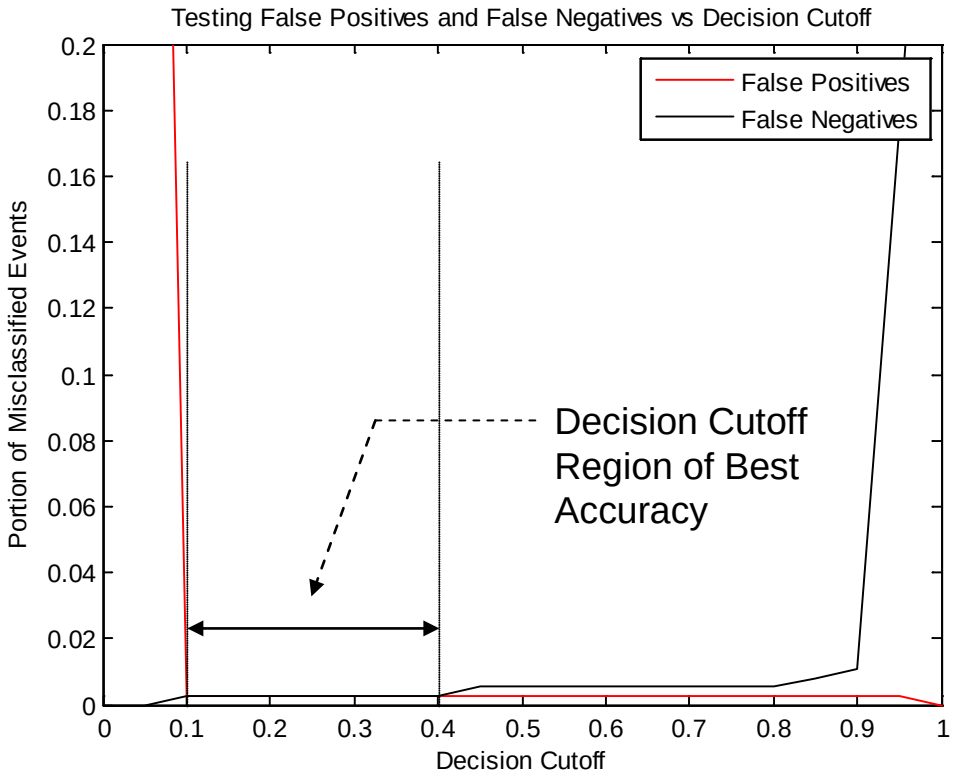

Figure 22. Accuracy of ANN Classifier with 100 dB Threshold

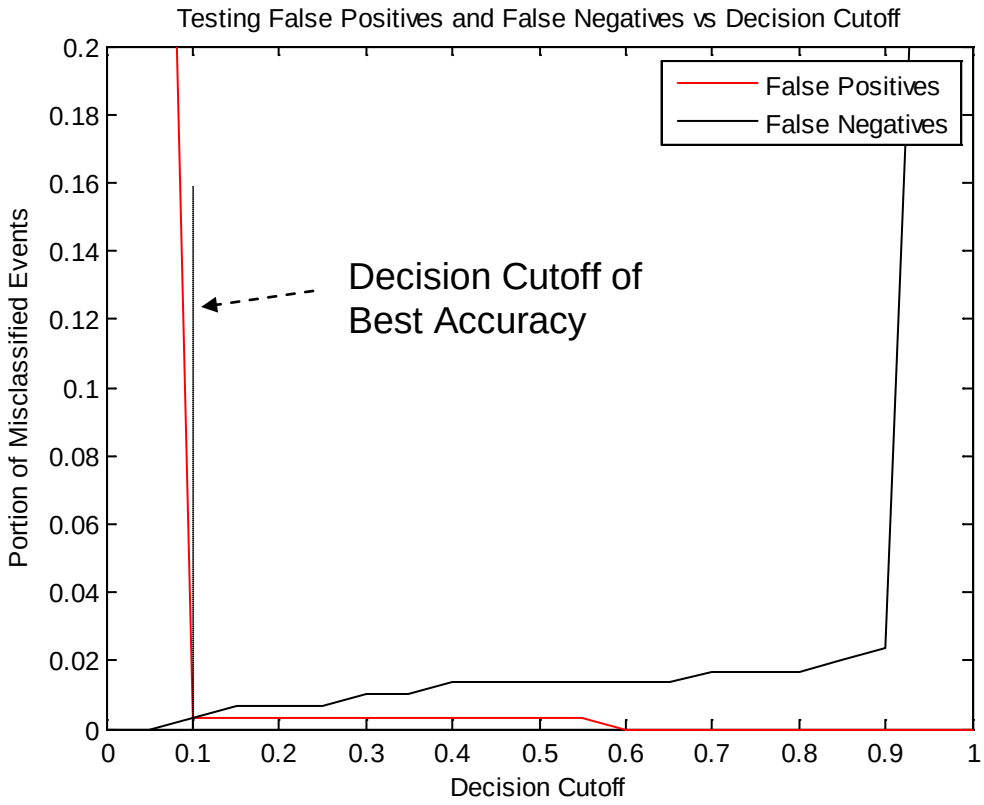

Figure 23. Accuracy of ANN Classifier with 105 dB Threshold

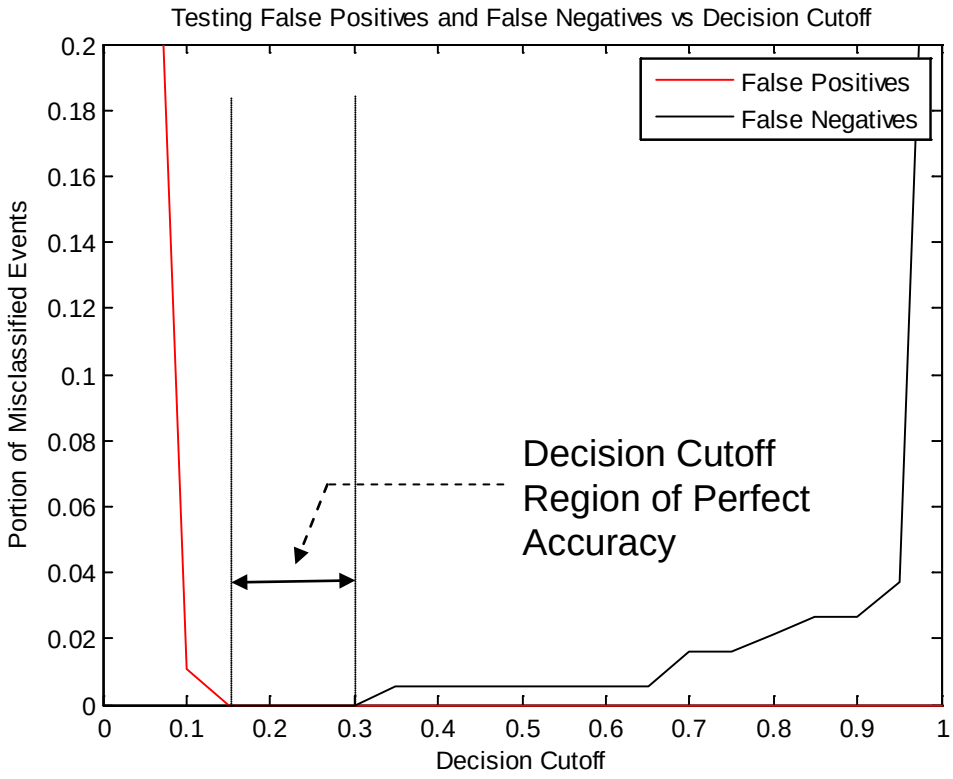

Figure 24. Accuracy of ANN Classifier with 110 dB Threshold

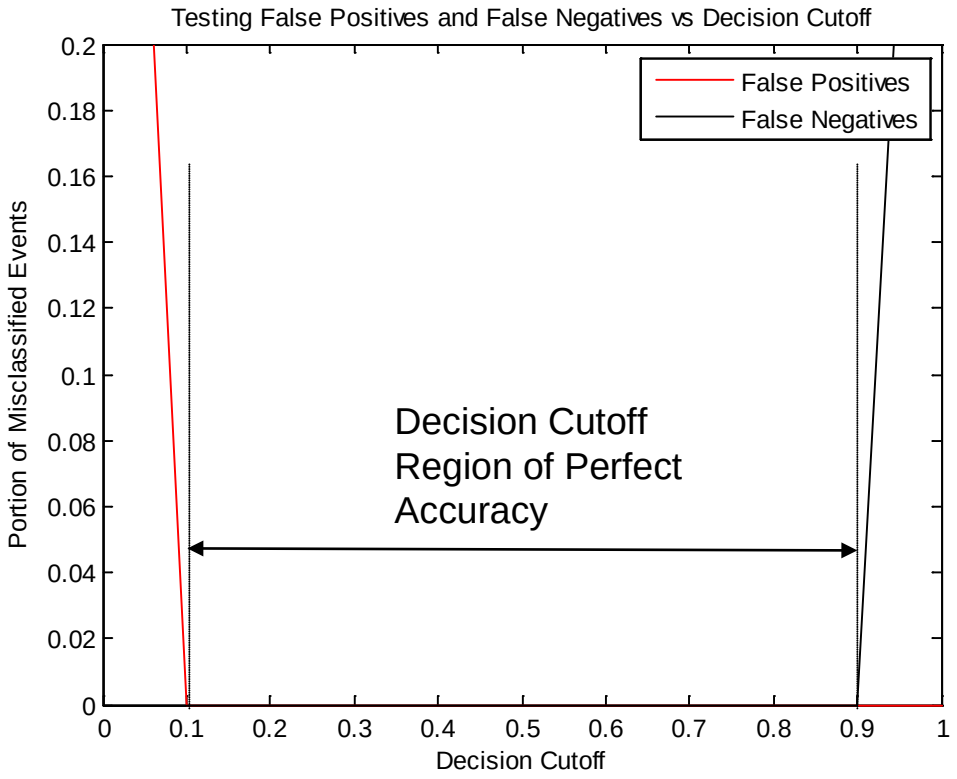

Figure 25. Accuracy of ANN Classifier with 115 dB Threshold

Table 3. Accuracy Results for Raw Data with Varying Threshold Levels

Threshold (dB)	Lowest FP Rate (%)	Lowest FN Rate (%)	Best Overall Accuracy (%)	Decision Cutoff Range for Best Overall Accuracy
None	0.3636	0.3636	99.09	0.80 – 0.85
100	0.2667	0.2667	99.47	0.10 – 0.40
105	0	0.3367	99.33	0.10
110	0	0	100	0.15 – 0.30
115	0	0	100	0.10 – 0.90

4.3.3 Decimation, Filtering, and Re-sampling Data Fidelity Analysis

When selecting hardware specifications, it was desired to modify the high-fidelity data to match less strict hardware requirements. To achieve this, the original data were digitally filtered using a 400 Hz low pass 4^{th} order Butterworth filter. Then the data were processed by decimating from 16 bits to 12 bits, and re-sampling at 1 kHz instead of the original 10 kHz. This decimated, filtered, and re-sampled (DFR) data were then used to train and test the ANN. Different L_{pk} thresholds were imposed on the data as well to decrease the effective dynamic range. The data set used for this analysis included all of the single channel data that were collected, which totaled 2263 waveforms. Two-thirds of the data were used as training data, and the remaining one-third of the data were used as test data. **Figure 26** – **Figure 30** show the results of this analysis. As can be seen in **Figure 26** and **Figure 27**, significant errors are incurred with low to no threshold applied to the data. This can be attributed to the fact that there will be a very low signal-to-noise ratio for these signals. As the threshold is increased however, the results become comparable to that of the unprocessed data. Therefore, this lower-fidelity data would be sufficient to execute the classifier successfully without significant loss of accuracy for thresholds of 105 dB or higher. In fact, 100% accuracy is obtained with a threshold of 110 dB applied to the data. **Figure 26** – **Figure 30** can be interpreted in exactly the same manner as described in Section 4.3.2, for **Figure 21** – **Figure 25**. **Table 4** summarizes the results from all of these figures.

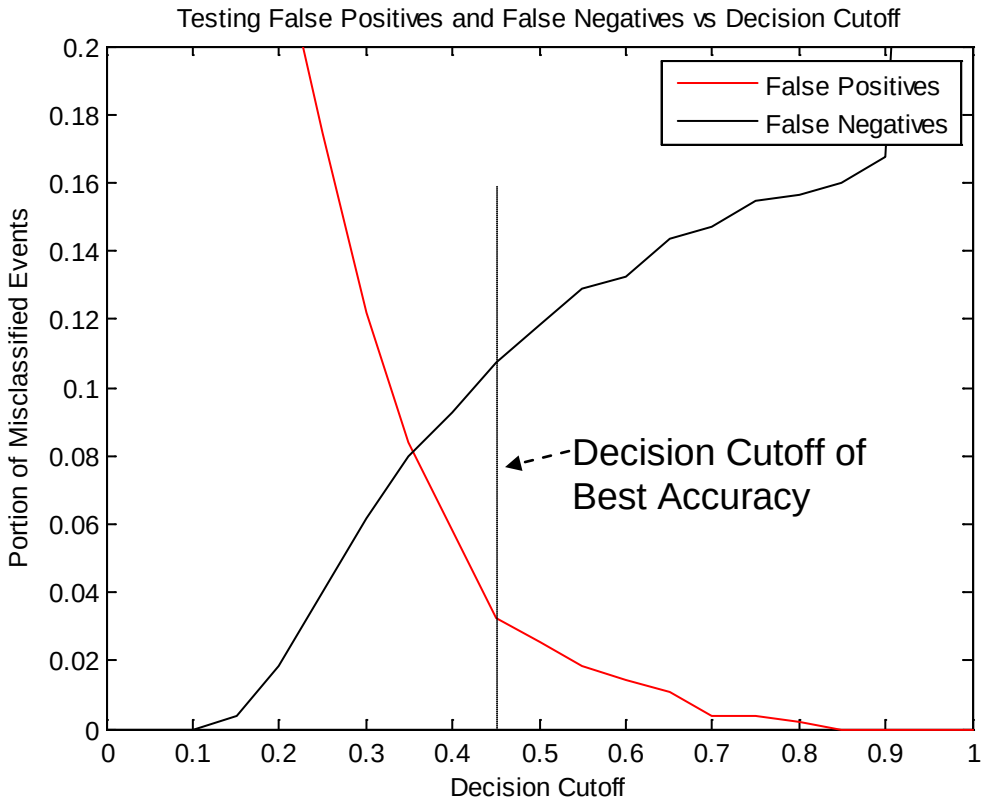

Figure 26. Accuracy of ANN Classifier with No Specified Threshold for DFR Data

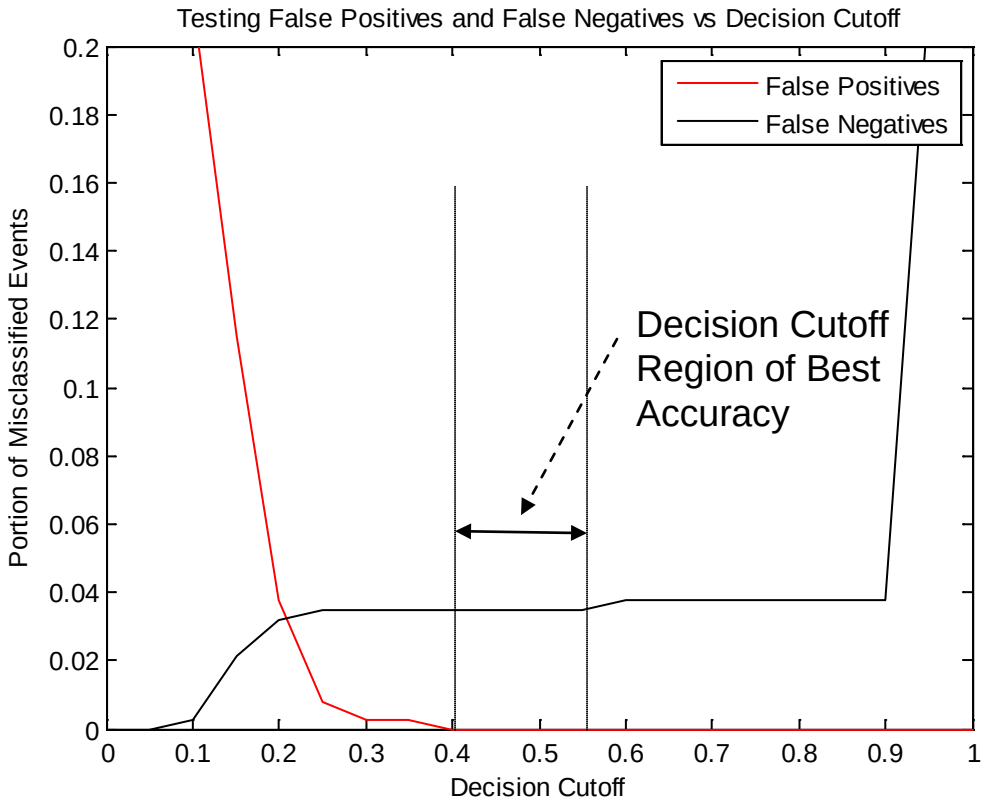

Figure 27. Accuracy of ANN Classifier with 100 dB Threshold for DFR Data

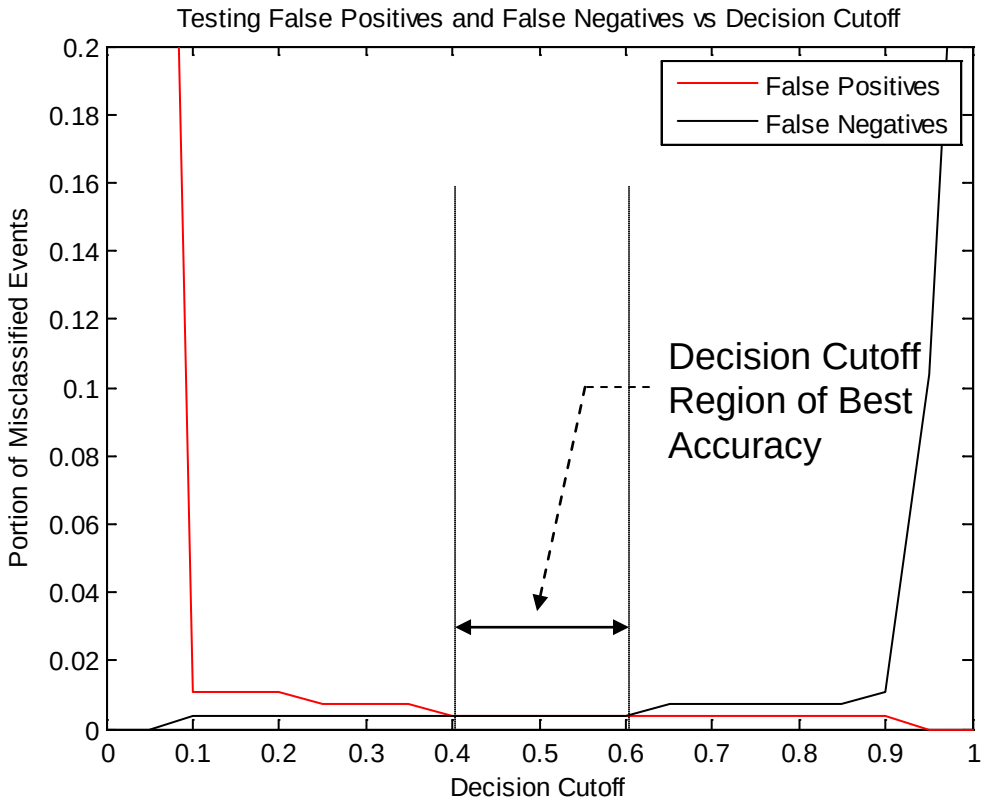

Figure 28. Accuracy of ANN Classifier with 105 dB Threshold for DFR Data

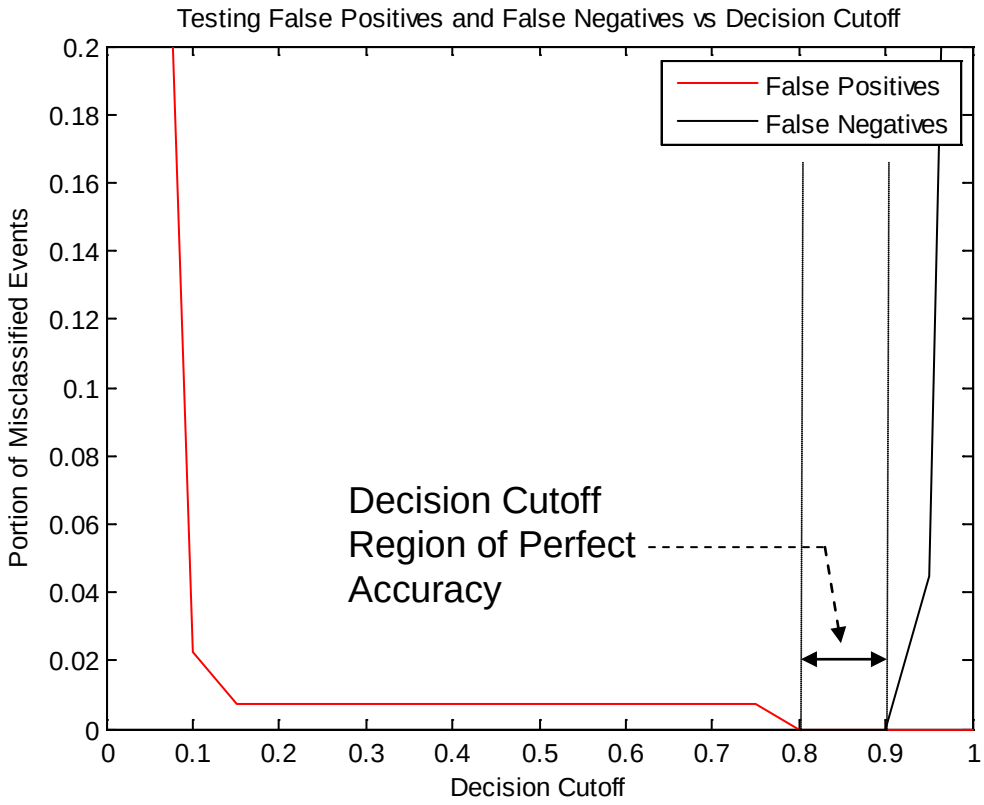

Figure 29. Accuracy of ANN Classifier with 110 dB Threshold for DFR Data

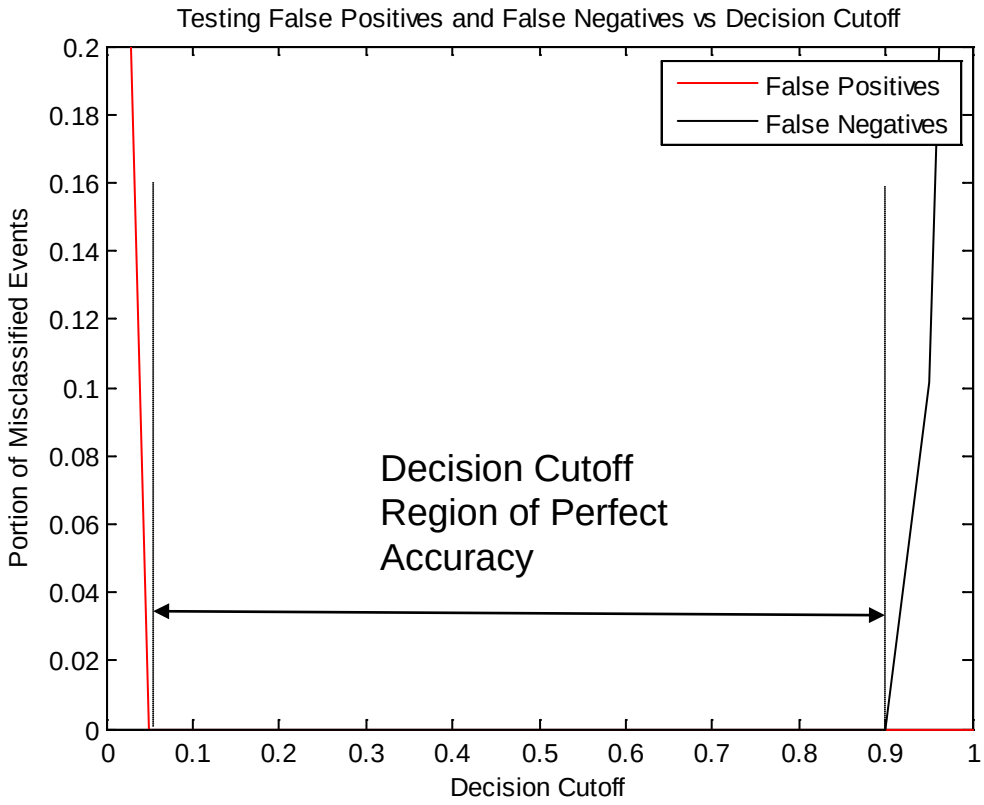

Figure 30. Accuracy of ANN Classifier with 115 dB Threshold for DFR Data

Table 4. Accuracy Results for Decimated Filtered and Re-Sampled (DFR) Data with Varying Thresholds

Threshold (dB)	Lowest FP Rate (%)	Lowest FN Rate (%)	Best Overall Accuracy (%)	Decision Cutoff Range for Best Overall Accuracy
None	8.0000	8.0000	86.00	0.45
100	0	3.2090	96.52	0.40 – 0.55
105	0.3571	0.3571	99.29	0.40 – 0.60
110	0	0	100	0.80 – 0.90
115	0	0	100	0.05 – 0.90

4.3.4 Determination of Low-Frequency Cutoff

To find out the effective frequency range for the classifier, it was desired to discern the lowest possible cutoff frequency for the classifier without loss of accuracy. An analysis was done by high pass filtering the data using a 2nd order high pass Butterworth filter. After filtering, the metrics were calculated from the data and passed into the ANN. Data from Base 3 were used as test data, since that particular base is subject to high winds speeds, which is the source of many false positive impulse detections. 184 wind and 43 impulse waveforms were used. The results of this analysis are summarized in **Figure 31**.

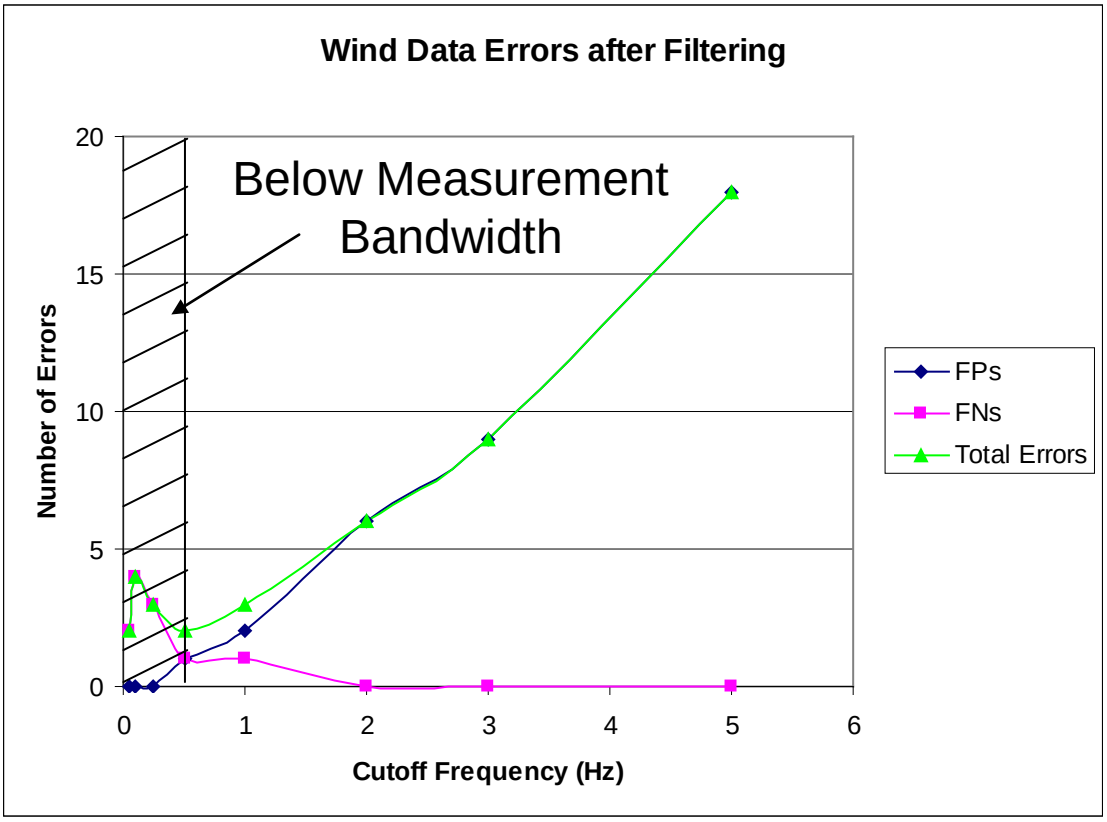

Figure 31. Classification Errors Due to High-Pass Filtering

The results of this study verified that good low-frequency response of the microphones is necessary for the classifier to work effectively. From **Figure 31**, the number of false positives (FPs) starts at zero, but then increases as the cutoff frequency of the filter is increased. There are also false negatives (FNs) that occur for lower cutoff frequencies (< 2 Hz), but this is not of concern, since the data acquisition was AC coupled with a low-frequency cutoff of a few tenths of a Hz. From the intersection of the FP and FN curve, it was determined that the actual lower bound on the bandwidth is 0.5 Hz.

Figure 32 shows an example wind waveform which helps to illustrate the importance of the low-frequency content of a signal. The solid blue line represents the original signal and the red dashed line represents the high pass filtered signal. The filter signal was obtained using a 2^{nd} order high pass Butterworth filter with a cutoff frequency of 2 Hz. Even though the shape of the waveform does not appear to change much except for some attenuation, the spectrum is affected significantly. The low frequency roll-off in the filtered spectrum makes it look similar to a spectrum for impulse noise. This effect is reflected in the metrics *m* and WSE. In fact, this example filtered waveform was misclassified by the classifier as impulse noise.

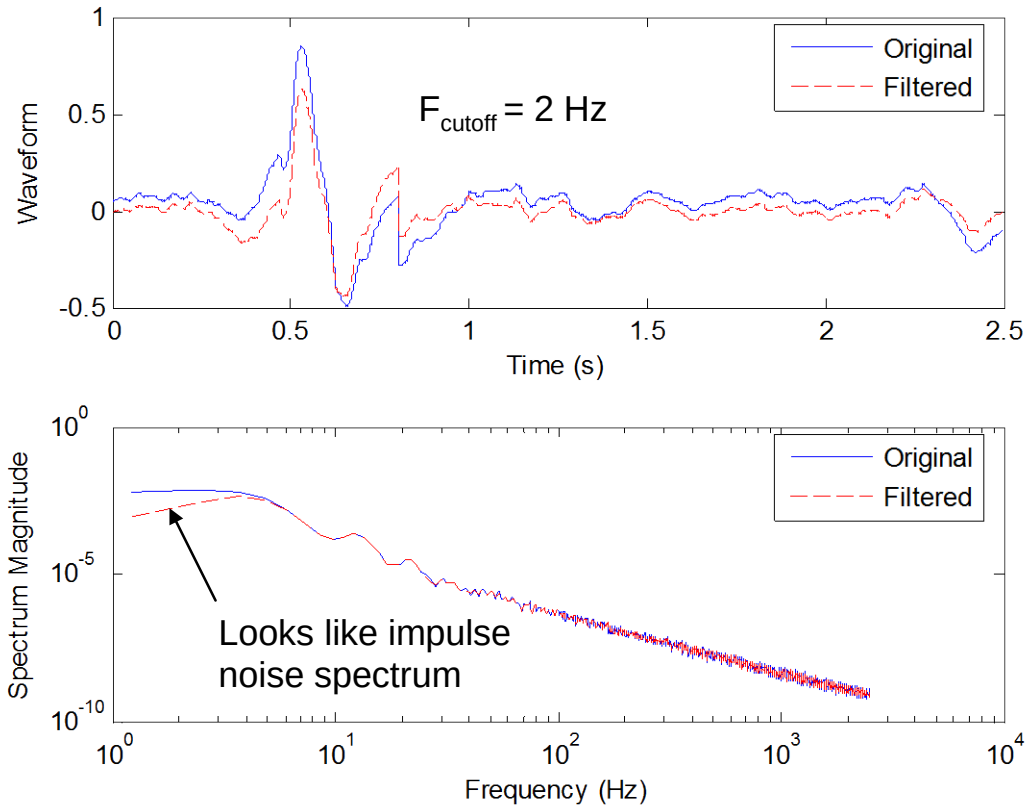

Figure 32. High-Pass Filtered (red) and Unfiltered (blue) Wind Noise (top-waveform, bottom-spectrum)

4.3.5 Generality of Classifier

The primary goal of this project is to obtain a classifier that is general enough such that it can be applied at any military base without loss of accuracy. To help to obtain this generality, data were collected at seven different military bases. In addition to this, variations in ordnance, weather, topography, and time of day were recorded. In order to test the generality of the classifier, the ANN was trained on all of the collected data except for the data from one particular base, which are used as the testing data. This process was repeated using data from each individual base as the testing data. This

analysis was done for single channel data. Since the data from Base 7 have multiple channels, only the channel obtained from the vertical microphone in the array was used. This particular microphone channel was selected because its orientation is the same as the single channel setup, and therefore would yield comparable output. The results of this are summarized in **Table 5**.

Table 5. ANN Classifier Accuracy Summary

Testing Location	Testing Max Accuracy (%)	Testing Max Accuracy (No FP) (%)	Training Max Accuracy (%)	Validation Max Accuracy (%)
Base 1	99.3	99.3	100	99.0
Base 2	99.8	99.3	99.6	98.4
Base 3	99.2	N/A	99.9	99.6
Base 4	99.0	N/A	100	99.4
Base 5	100	100	99.9	99.1
Base 6	100	100	100	99.0
Base 7	100	100	99.5	99.2

As can be seen in **Table 5**, all of the accuracies for the classifier are very high ($\geq$ 99.0%). The testing max accuracies are important because they represent how well the classifier would perform on any "blind" data. "Blind" data are data that are not "seen" by the network, *i.e.*, they have no effect on the training process. In the case of a real time classifier, the input data will be "blind," since they are new to the network and will be able to output an accurate result without the need for training.

5.0 MICROPHONE ARRAY METHODS

In order to further improve the characterization capabilities of military noise monitoring stations, a microphone array was used to collect four simultaneous channels of data. The geometry of this array is outlined in section 3.0 of this thesis. It has been shown that multiple channel data can be effective in the rejection of wind noise (Benson, 1996). The multiple channel data allow for a more detailed analysis of each noise trigger event. The additional degrees of freedom provided by the microphone array allow for correlation analysis as well as sound source localization to determine the direction of the event source. This additional information will help to further characterize a given input signal, thus adding more classification capabilities (Rhudy *et al.*, 2009).

5.1 MIN PEAK CORRELATION COEFFICIENT

Sound waves will propagate through fluids as longitudinal waves (McCowan, 2001). Making a far-field approximation, it is assumed that a military impulse event will propagate across the microphone array as a uniform wave front. Each microphone will individually measure the impulse event, differing from the other microphone channels by a time delay (Bendat and Piersol, 1993). By calculating the cross-correlation function for each pair of microphones, these time delays can be found. Given the speed of sound in air and the geometry of the array, the maximum possible time delay between the

microphone channels, τ_{max}, was conservatively estimated to be 12 ms. If the time delay, τ, between the channels satisfies $|\tau| \geq \tau_{max}$, the event can immediately be rejected as an uncorrelated event, such as wind (Rhudy *et al.*, 2009).

In addition to this uncorrelated noise rejection capability, the cross-correlation coefficient functions can also be used as a measure of how correlated the overall event is. Considering only values of τ such that $|\tau| < \tau_{max}$, the peak values of the cross-correlation coefficient functions are calculated. The minimum of these values, ρ_m is then taken to obtain one overall scalar measure of the correlation of the event, called the *min peak correlation coefficient*. The *min peak correlation coefficient* satisfies the inequality, $-1 \leq \rho_m \leq 1$, where larger values of ρ_m represent a more correlated waveform. To illustrate the effectiveness of this value, a histogram was plotted in **Figure 33** for the wind and Bradley data that was collected. Bradley waveforms with L_{pk} values less than 100 dB were not considered.

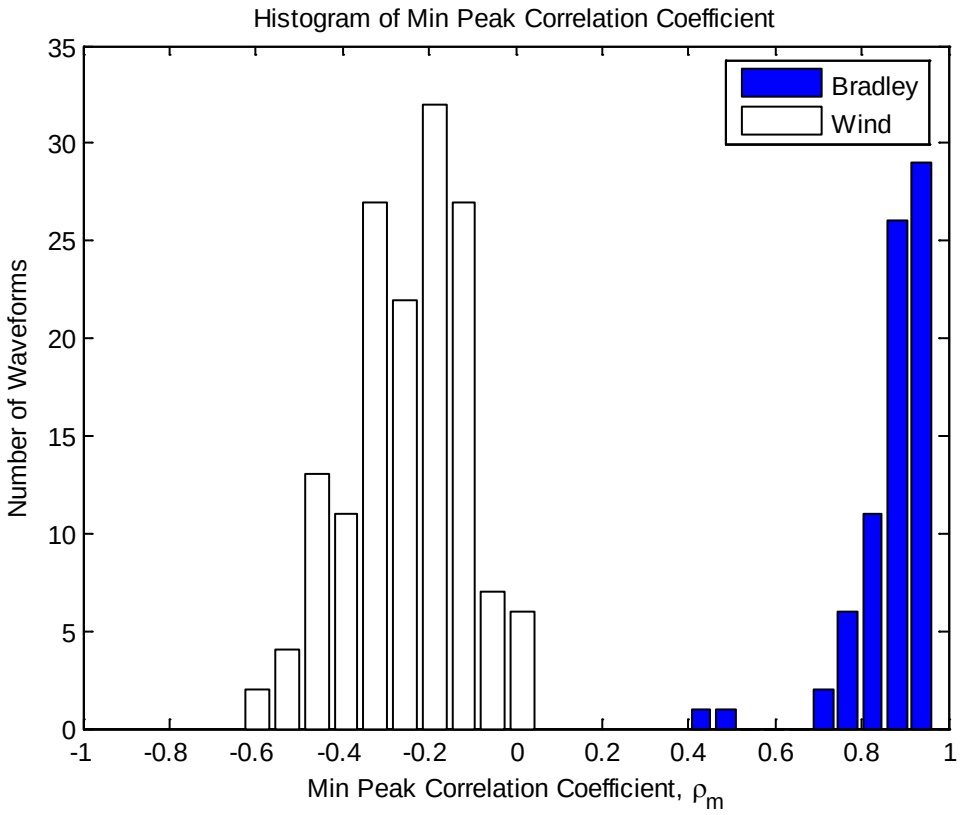

Figure 33. Histogram of Min Peak Correlation Coefficient

Figure 33 shows a clear distinction between values of ρ_m for wind and Bradley waveforms. As predicted, Bradley waveforms have higher values of ρ_m which indicates that they are more correlated among channels than the wind. There are two Bradley waveforms that received relatively lower values of ρ_m = 0.4 and 0.5. These recordings contain significant amounts of wind noise in addition to the Bradley events. This therefore means that the channels are less correlated with each other, which is expected (Rhudy *et al.*, 2009).

5.2 SOUND SOURCE LOCALIZATION

Another benefit of the microphone array is added degrees of freedom which can be used to determine the direction of the event source. Assuming that the array is far from the source of the event (far-field approximation), a military impulse event will propagate across the array as a uniform wave front, which is illustrated in **Figure 34**.

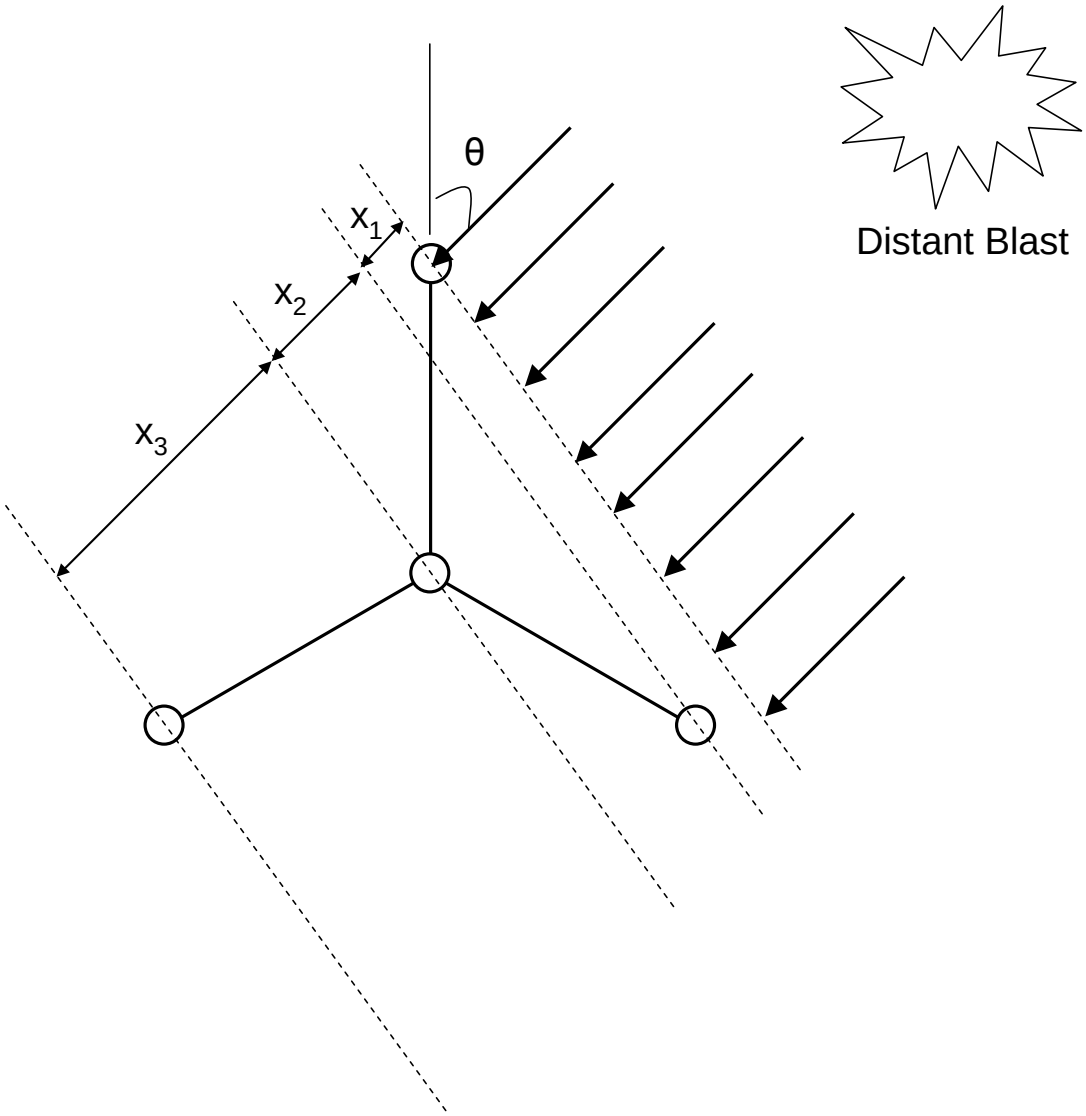

Figure 34. Top-View of Microphone Array Setup

By using cross-correlations to calculate the time-delays between the signals, the angle of incidence, θ, of the wave front can be found. These time delays are related to the distances between the microphones by using the simple relation

$$x = c \cdot \tau \tag{20}$$

where x is the distance between the microphones along the direction of θ, c is the speed of sound in air, and τ is the calculated time delay between the microphones. θ is calculated using

$$\theta = \cos^{-1}(c \cdot \tau_{lead}) \tag{21}$$

where τ_{lead} represents the time delay between the vertical microphone and the leading microphone, which is the first to read the impulse signal. Using the order in which each microphone read the impulse signal, the sign of θ is determined. θ points in the direction of the sound event source, which is measured relative to the direction of the leading microphone. The resolution of the calculation of θ will be limited primarily by the sampling rate, since this determines the resolution of the time delay. Increasing the sampling rate will increase the resolution of θ. A sampling rate of 10 kHz will yield a maximum of a 50 μs error in the time delay. Due to the non-linearity of the cosine function, the potential error in the calculation of θ depends on the value of τ_{lead}. Assuming that c = 343 m/s, a change is time delay of 50 μs can lead to errors in θ ranging from 1.2° to 6.3°, with a mean error of 1.9°. Temperature is another factor that can affect

this resolution, due to its effect on *c* (Rhudy *et al.*, 2009). This temperature dependence is given by the equation,

$$c = \sqrt{\gamma R T_k} \tag{22}$$

where γ is the ratio of specific heats, R is the gas constant (R = 0.287 kJ kg^{-1} K^{-1} in air), and c is the speed of sound at the absolute temperature T_k. This relation assumes that air can be modeled as an ideal gas (Norton, 2003). Note that γ is also a function of temperature, but does not change significantly as temperature changes.

During one of the recording sessions at Base 7, 40 Bradley recordings were taken from a single approximate firing location. Data were recorded at one stationary position that was approximately due north from the firing location. A histogram of the calculated values of θ is presented in **Figure 35**. The values of θ calculated from this data are measured clockwise relative to the microphone facing approximately southeast. In an absolute sense, these angles ranged from -1.3° to 8.1° counterclockwise from south. This is a reasonable result since the exact source of the impulse waves is unknown, and would typically consist of multiple point sources spread across an area. A plot of this range overlaid on a top-view of the microphone array is given in **Figure 36** (Rhudy *et al.*, 2009).

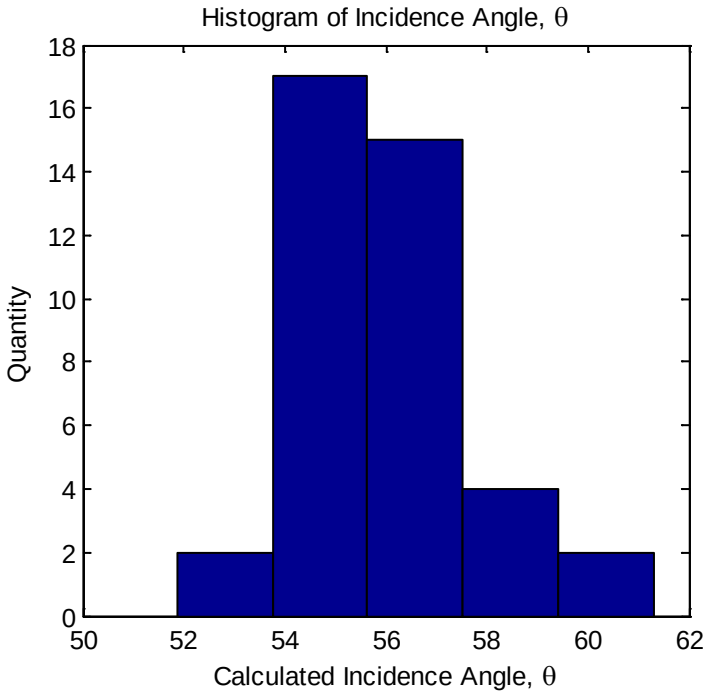

Figure 35. Histogram of Calculated Incidence Angle

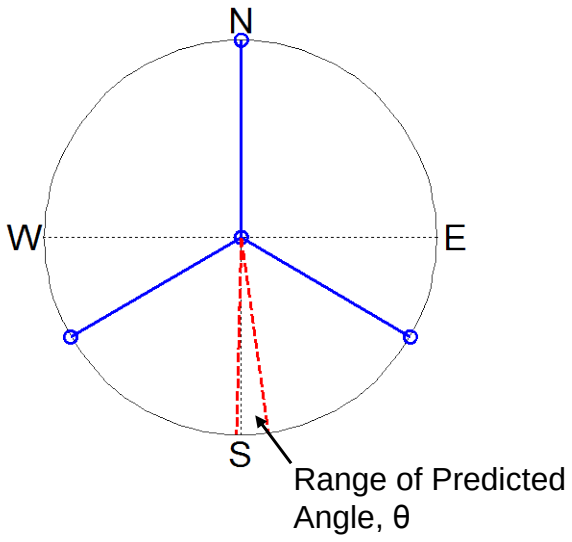

Figure 36. Plot of Range of Predicted Value

5.3 CORRELATED SIGNAL SYNTHESIS

Due to the great success of previous work on single channel data (Bucci and Vipperman, 2006, 2007, Bucci, 2007), a method was established to combine the four channels of microphone data into one correlated signal. The first step in combining the four signals is to shift the signals by the proper time delay, τ, which was calculated previously through cross-correlations. This will align the signals such that a military impulse event present in the signals will occur at the same time. Once the signals are aligned, they must be synthesized together in such a way that the correlated portion of the waveform is emphasized, while the effects of uncorrelated noise are minimized. To accomplish this goal, a weighted average is taken at each point in time to obtain one overall correlated signal, $p_{corr}(t)$. Since wind noise is a turbulent or random process (Morgan and Raspet, 1992), it will not present itself equally across all four channels. Therefore, if a signal differs greatly from the other signals at a particular point in time, this value is considered undesired noise, and should be given a small weight. However, channels that are closely correlated should be given higher weights, since they represent correlated sound. To calculate the weights, the following equations are used:

$$E_i(t) = \sum_{j=1}^{4} \left| \frac{p_i(t) - p_j(t)}{p_j(t)} \right|, \qquad i = 1, 2, 3, 4 \tag{23}$$

$$W_i(t) = 1 - \frac{E_i(t)}{\sum_{j=1}^{4} E_j(t)}, \qquad i = 1, 2, 3, 4 \tag{24}$$

(25) $$w_i(t) = \frac{W_i(t)}{\sum_{j=1}^{4} W_j(t)}, \qquad i = 1, 2, 3, 4$$

where i = 1, 2, 3, 4, represents microphone channel i, $p_i(t)$ is the sound pressure, $E_i(t)$ is the relative signal error, $W_i(t)$ is the un-normalized weight, and $w_i(t)$ is the normalized weight of channel i. To calculate the overall correlated sound pressure, $p_{corr}(t)$, a weighted average is taken at each point in time using the normalized weights, $w_i(t)$, such that

(26) $$p_{corr}(t) = \sum_{i=1}^{4} w_i(t)\, p_i(t)$$

To illustrate this synthesis method, **Figure 37** – **Figure 40** represent the sound pressure from microphones 1 – 4 respectively for a particular Bradley waveform. In this example, there is significant wind noise present in microphone 4 (**Figure 40**) due to an improper fit of the windscreen, and there are also some noise spikes in microphone 3 (**Figure 39**) due to light rain, *e.g.*, at approximately 1.2, 1.3, 1.6, and 1.8 seconds (Rhudy *et al.*, 2009).

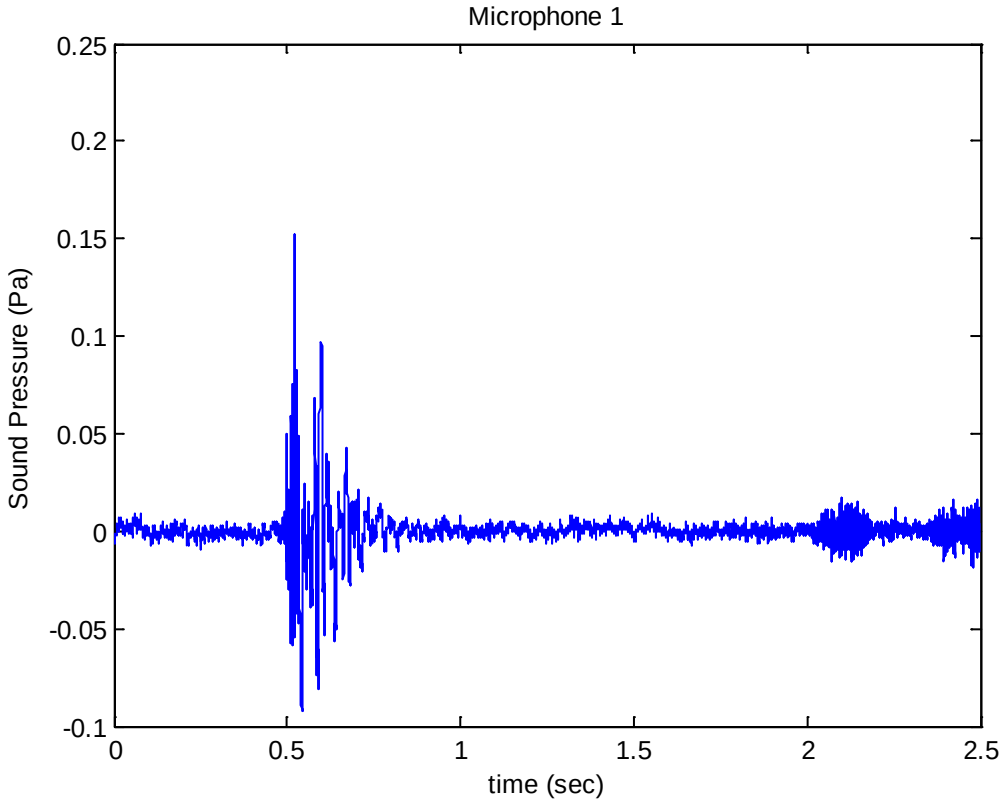

Figure 37. Sound Pressure of Microphone 1

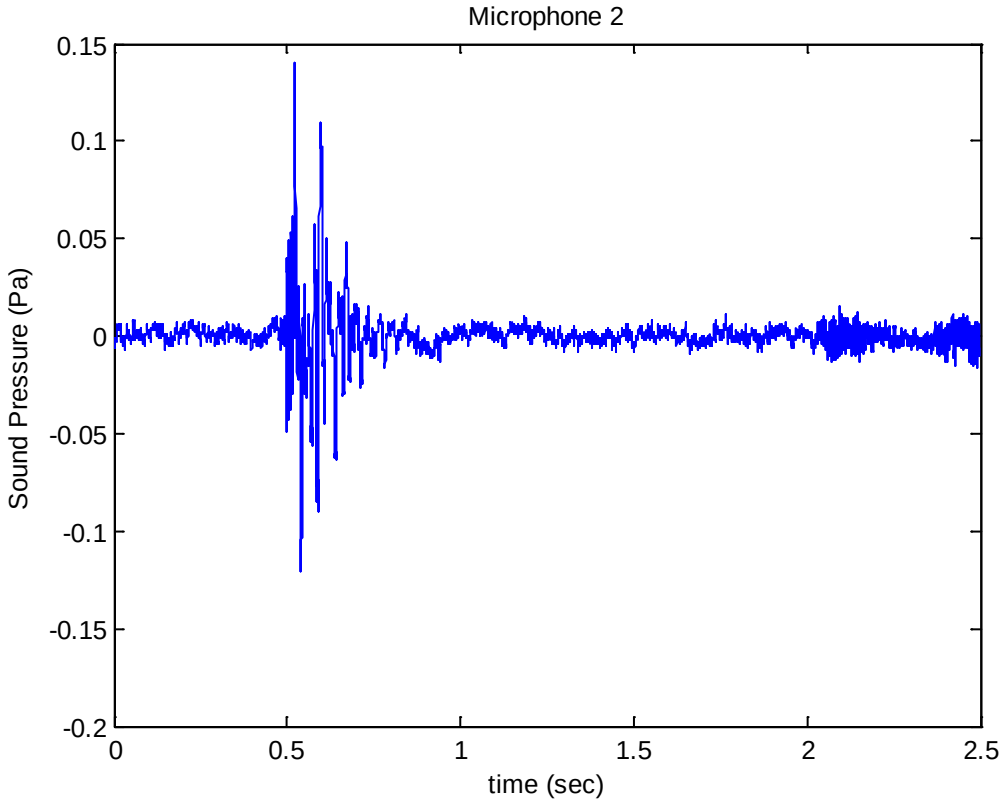

Figure 38. Sound Pressure of Microphone 2

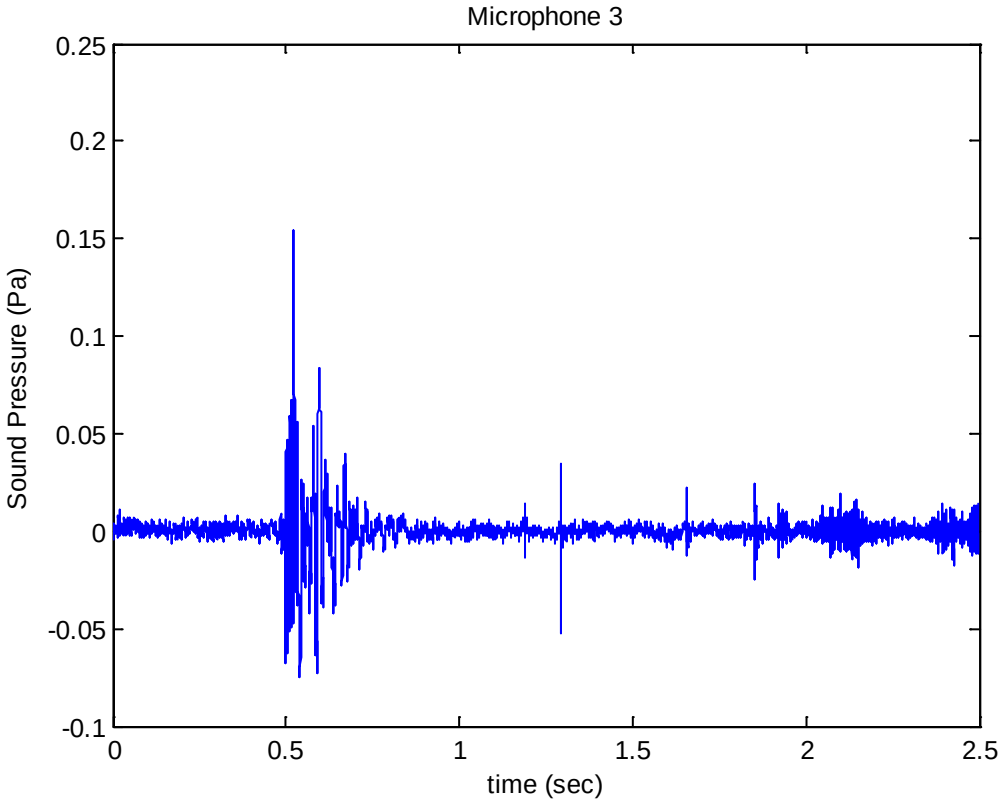

Figure 39. Sound Pressure of Microphone 3

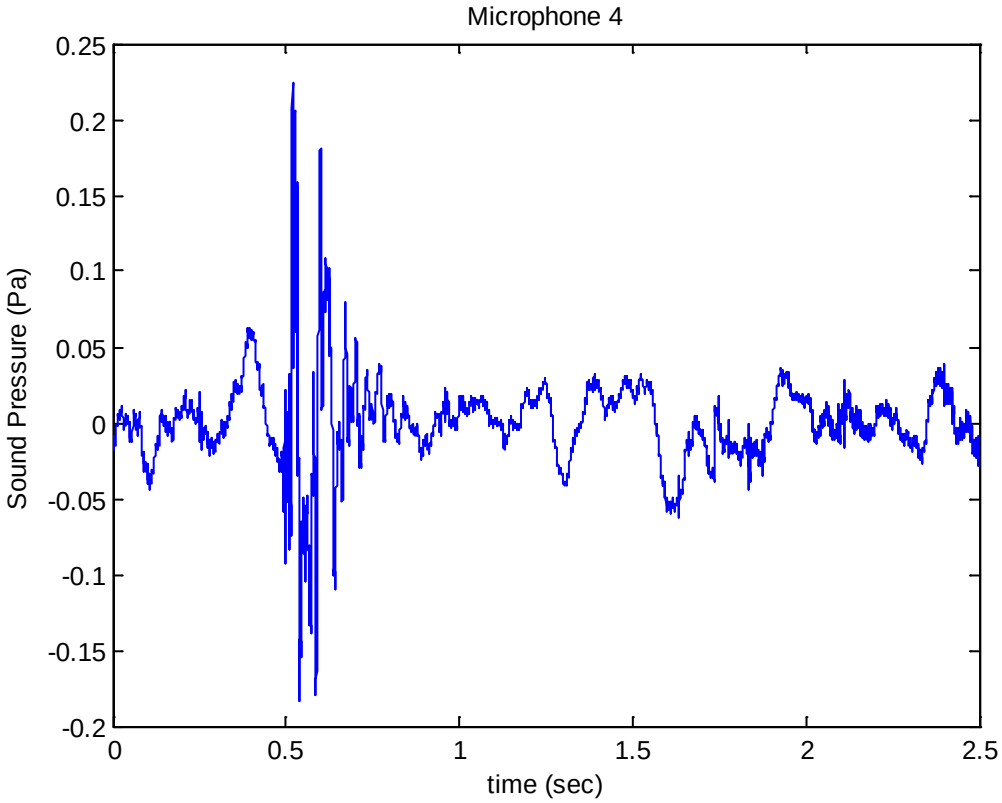

Figure 40. Sound Pressure of Microphone 4

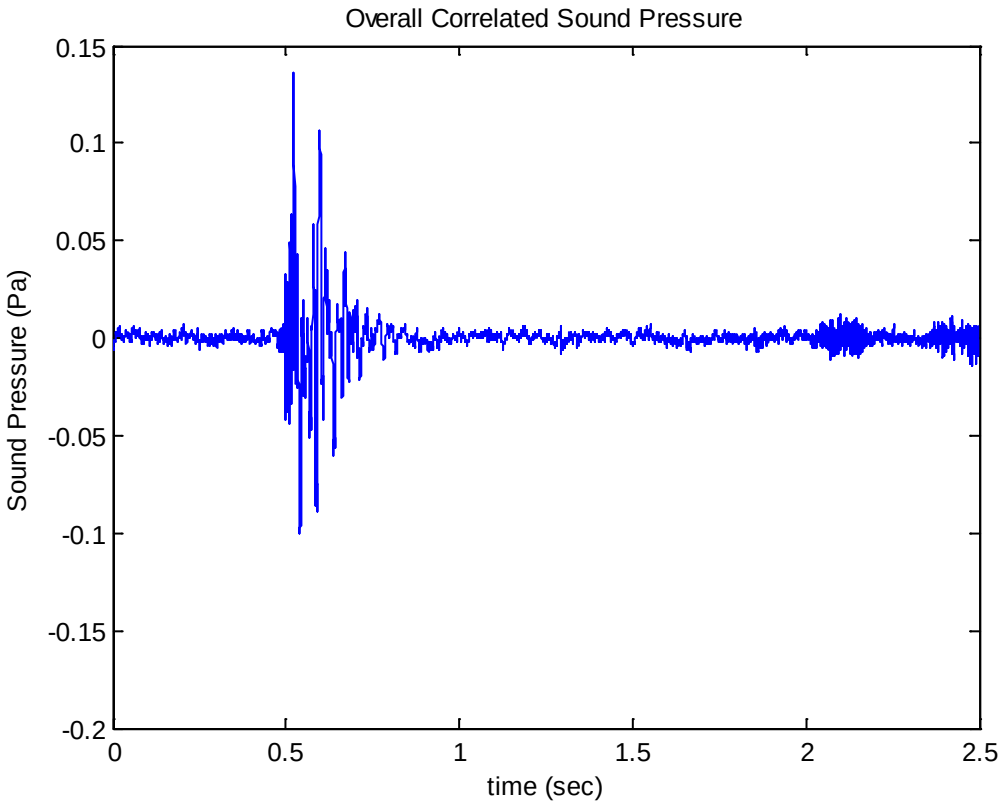

Figure 41. Overall Correlated Sound Pressure

The overall correlated sound pressure seen in **Figure 41** provides a clear representation of the military impulse event, without uncorrelated noise in the signal. From this example, it can be seen that this weighted average synthesis method can be effective in minimizing the amount of uncorrelated noise, such as wind, in the signal.

6.0 REAL TIME IMPLEMENTATION

6.1 SOFTWARE DEVELOPMENT

All of the classification algorithms were initially developed using the MATLAB programming language created by MathWorks™. MATLAB was selected for this development due to its ease of use and vast library of built-in signal processing functions. Another advantage is MATLAB's easy handling of vectors and matrices. Previous work by Bucci and Vipperman took advantage of MATLAB's neural network toolbox, which provides many capabilities in training and implementing neural networks (Bucci and Vipperman, 2006, 2007, Bucci, 2007). In order to create a real time implementation, the algorithms needed to be written in a compiled programming language. C was therefore selected for further development. Before developing codes directly for a digital signal processing (DSP) board, C codes based on the original MATLAB codes were written to run on a PC. This process was done in order to ensure the accuracy of the C implementation before beginning DSP development. This C implementation includes the calculation of the four scalar metrics and using them to find the output of the ANN. In order to calculate the frequency metrics, a Discrete Fourier Transform (DFT) is required. While this is a built in function in MATLAB, an implementation had to be found in C. The implementation that was initially used was found in *Numerical Recipes in C* (Press,

1997). A test data set of 1,615 measured waveforms was used to verify the accuracy of the C implementation versus the MATLAB implementation for a given trained ANN. Both the C and MATLAB codes accept each of the 1,615 waveforms as an input, calculates the four scalar metrics, and passes those metrics into the ANN classifier. The differences between the classifier outputs from each programming language can be seen in **Figure 42**. This shows that the C code closely matches that of MATLAB, with negligible differences in output.

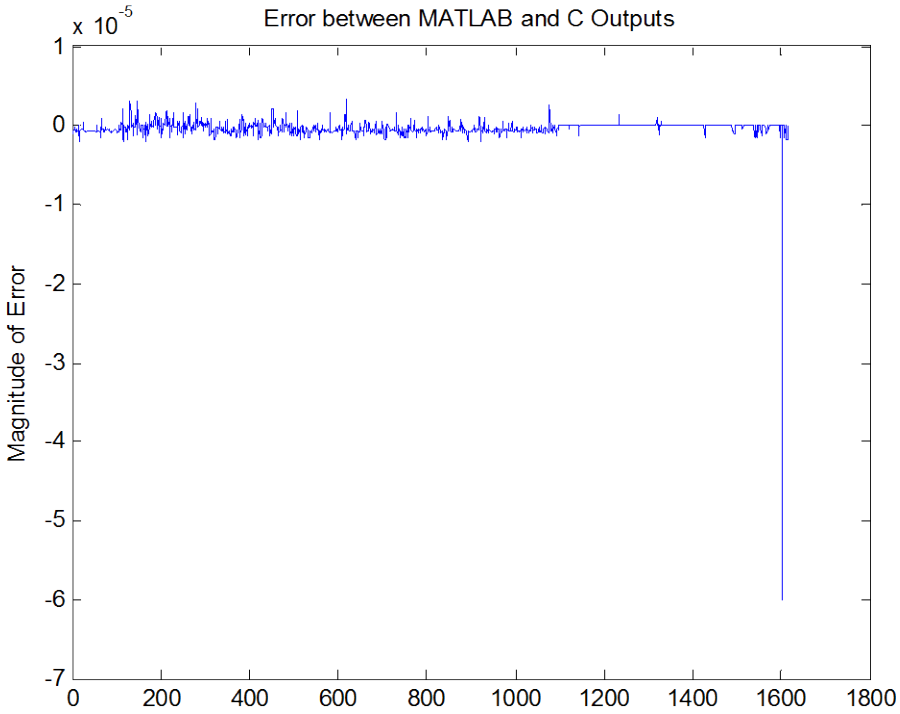

Figure 42. Error Between Classifiers Implemented in C and MATLAB

Due to the complicated nature of the ANN training procedures, all network training was done in MATLAB. Once the ANN is trained on a sufficient amount of data,

it should not require any further training. Therefore the ANN can simply be implemented on a DSP board, without having to worry about retraining. If ANN training in C is later found to be necessary, the Fast Artificial Neural Network Library (FANN) was found to be comparable to MATLAB in training and implementing an ANN (Fast Artificial Neural Network Library).

Although a fully functional classifier was written in C, this implementation was not optimized for execution on a DSP board. The Integrated Performance Primitives (IPP) library by Intel was used to aid in the development of the specialized code for the DSP board. This library provided various signal processing functions. These functions can operate using either fixed point or floating point computations. Since the hardware utilizes a fixed point processor, fixed point computations were used whenever possible. The remaining calculations required floating point computations due to accuracy and scaling issues. These floating point computations run on the fixed point processor by using fixed point emulation, which requires additional run time.

Due to run-time restrictions, the real time system sampling rate was selected to be 5 kHz instead of the original 10 kHz. This new sampling rate was not found to impact the results negatively. Data is collected over a one second window. Each resulting waveform to be processed contains 5,000 data points. The original collected waveforms were re-sampled to match this specification and tested for accuracy. Using the new waveforms, the four scalar metrics were calculated using a C implementation which closely mirrors the DSP implementation. These metrics were then used to train an ANN. A network was obtained with 100% training, validation, and testing accuracy. This network was used in a real time DSP implementation.

In order to calculate the frequency metrics, a Fast Fourier Transform (FFT) was required. It was desired to perform this operation using fixed point computations, since it requires significant run-time to run using floating point. The IPP library FFT did not run properly on the selected DSP system, so the Kiss FFT was selected instead. The Kiss FFT is a small, simple, mixed radix FFT library that can run using either fixed point or floating point computations (Source Forge). The real time implementation uses a single 8,192 point FFT to calculate the two frequency metrics, WSE and *m*. Since each event consists of only 5,000 data points, the data is padded with 3,192 zeros to perform the 8,192 point FFT operation. Due to the changes in sampling rate and FFT resolution from the original design, the calculations for WSE and *m* had to be adjusted accordingly. The desired frequencies to be investigated are the low frequencies up to 100 Hz, not including 0 Hz. For the sampling rate of 5 kHz, and FFT resolution of 8,192 points, there are 164 frequency bins in the range of 0.6 Hz – 100 Hz. The fit for *m* is conducted across these bins, and WSE is calculated from the first 163 of these bins, using the same basic formulas as defined in section 4.3.1.

In addition to the ANN classification algorithm, the Impulse Noise Bearing and Amplitude Measurement and Analysis System (BAMAS) developed by Applied Physical Sciences Corp. (APS) is used (Abraham, 2005). This system calculates a direction of arrival (DOA) or bearing at each individual setup, and later combines these bearings whenever possible to calculate the position of an event, *i.e.*, whenever multiple arrays record the same event. In addition to these calculations, the BAMAS rejects uncorrelated noise sources, such as wind, and otherwise combines the four microphone channels into a single channel. The single channel output from the BAMAS can be used to calculate the

scalar metrics to be used as inputs for the ANN classifier. The system was designed so that it can be tested using both the BAMAS and ANN classifier combined, as well as implementing each classifier individually.

6.2 HARDWARE DESCRIPTION

In order to execute the classifier algorithm, a hardware platform needed to be selected. Since the classifier algorithm was to be implemented on a digital signal processing (DSP) board, the approximate number of computations was determined. The number of floating point operations (FLOPs) was found for the classifier algorithm based on the number of data points, *n*, in a given waveform. For 5,000 data point waveforms, the total number of FLOPs was estimated to be 146,004. A breakdown of these computations is given in **Table 6**. Since this was an estimate of the number of FLOPs, the actual number of computations was conservatively estimated to be 0.3 MFLOP. In addition to this estimate, there will also be overhead due to the fixed point emulation that will be executed on the DSP board.

Table 6. Number of Computational Operations in Classifier

Type of Operation	**Number of Operations**	**Number of Operations for n = 5,000**
Addition/Subtraction	$9 \cdot n + 739$	45,739
Multiplication/Division	$14 \cdot n + 540$	70,540
Exponential	270	270
Hyperbolic Tangent	2,560	2,560
Base 10 Logarithm	8,200	8,200
Square Root	200	200
Fast Fourier Transform (FFT)	$n \cdot \log(n)$	18,495
Total:		**146,004**

The following hardware platform was specified by Applied Physical Sciences Corp. (APS) in collaboration with the University of Pittsburgh. The selected DSP board is a PC104 single board computer (SBC) that was developed by Arcom. This board uses an Intel XScale PXA255, which is a 400 MHz advanced reduced instruction set computer (RISC) machine (ARM) compliant central processing unit (CPU). This processor was chosen because of its small power dissipation of only 2W for typical operation, and its high CPU speed which allows for real time implementation of the classification algorithm (Arcom).

The data acquisition (DAQ) board selected to interface with the PXA255 is the STX104 board developed by Apex Embedded Systems. This board provides 16 channels of analog input, and 2 channels of analog output with 16-bit resolution. One of the benefits of this board is its high reliability even in harsh environments. This is an important feature due to the desired outdoor application at military bases. The inputs can

sample up to 200 kHz, which is well above the necessary 5 kHz for the classification algorithm (Apex Embedded Systems).

To power the DSP board and the data acquisition board, Tri-M Engineering's HESC104 60 Watt High Efficiency Uninterruptible Power Supply (UPS) PC104 was selected. ±5V and ±12V can be supplied. The HESC104 provides advanced power management functions and smart battery charging due to its flash based microcontroller. For example, the HESC104 can be programmed to power off the main outputs in 60 seconds and then turn them on again 12 hours later. This particular power supply accepts Smart Management Bus (SMBus) compatible batteries which allows for battery-controlled charging (Tri-M Engineering).

One individual PC/104 stack unit consists of the Arcom PXA255 SBC, the STX104 DAQ board, the HESC104 UPS, a custom signal conditioning board, a PC/104 ISA extension card, and a rechargeable Lithium-Ion battery. A picture of one of these units is shown in **Figure 43**.

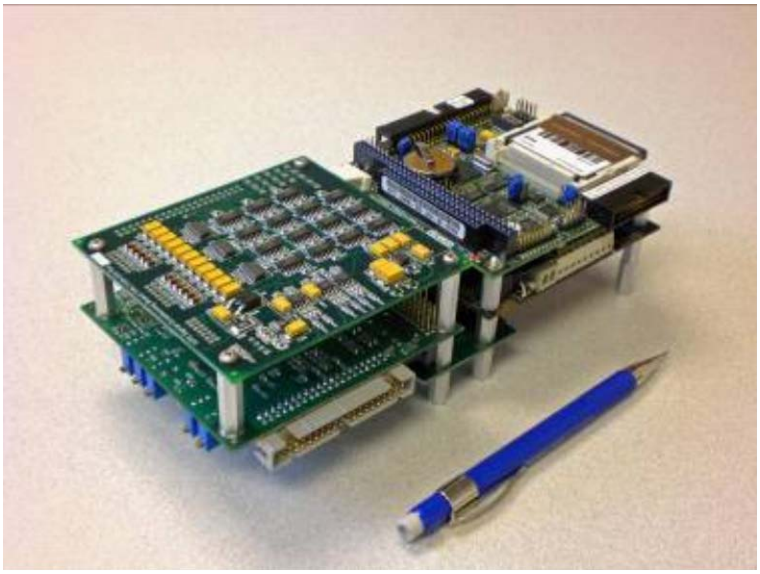

Figure 43. PC/104 Stack with Low Power 400MHz CPU, 200kHz DAQ, Power Supply, and Custom Signal Conditioning Board

Each of these units is able to detect and store blast information locally. The individual units also communicate wirelessly with the base station computer using PC/104 compliant 900MHz Serial Port or Ethernet bridges. These can transmit and receive over long distances approaching 20 miles using Yagi-directional antennas allowing the system to monitor large areas. A block diagram of this overall system is displayed in **Figure 44**.

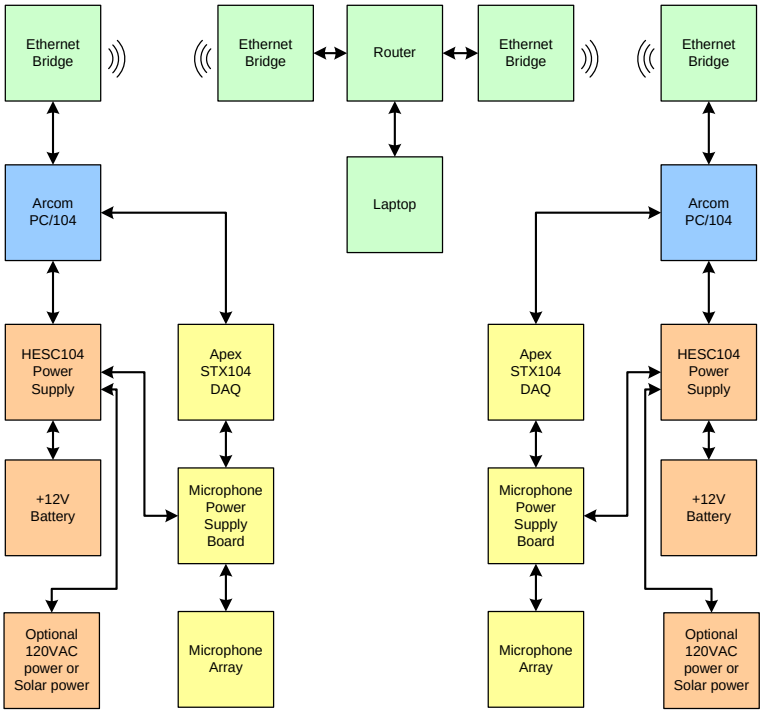

Figure 44. Block Diagram of BAMAS System Configuration

Once data is sent from a unit to the central base computer, it can be combined with data from other units by triangulation and logged to disk. This log file can be imported into Google Earth using an interface that was developed by Applied Physical Sciences Corp. (APS). This website catalogues all events detected by the Bearing and Amplitude Measurement and Analysis System (BAMAS) and projects this information onto a satellite map using an estimate of the source latitude and longitude. Information such as time, peak level, horizontal, vertical angle, and file name may be displayed when the user clicks on the projected way point. To push detection information to the website, the base station first creates an email, in extensible markup language (XML) format, which is sent to a designated email account. The website scans this email account for recent messages, and posts them in the table above the satellite map. By clicking on one of the

messages in the table, the website will automatically update the satellite map to display the waypoint and information associated with that event. The user also has the option to "view all" in which case the map will display all waypoints. **Figure 45** shows an example screen of results from this Google Earth interface.

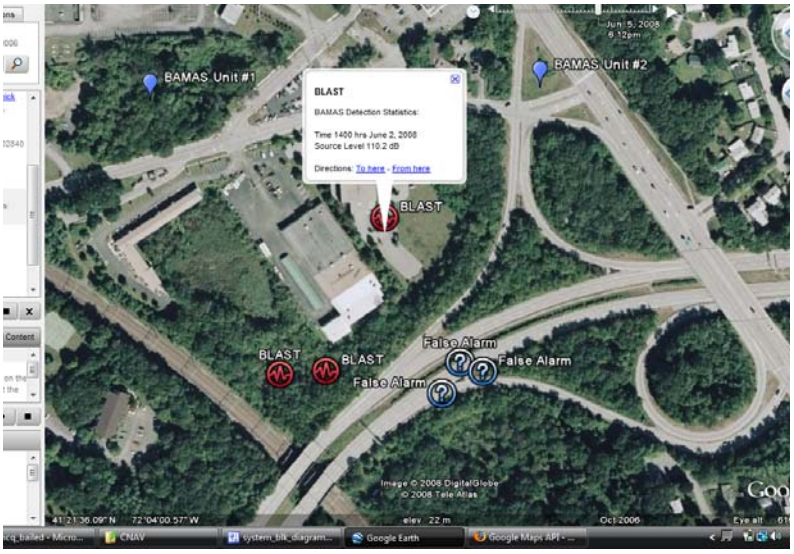

Figure 45. Detected Events Cataloged by BAMAS System Displayed in Google Earth

6.3 FIELD TESTING

6.3.1 Prototype System

In order to evaluate the performance of the system, two prototype units were created by Applied Physical Sciences Corp. (APS), and installed at Base 1. Each unit includes a microphone array with four Knowles BL-7242 microphones, as well as an additional fifth

ACO 7052 microphone that possesses the low-frequency response necessary for the ANN classifier. Each microphone is protected by a high-quality spherical windscreen. The microphones connect to a large metal box, which contains the PC/104 stack and battery. The battery is recharged by a solar panel. An anemometer was installed with each unit to monitor and report wind speed and direction. Each unit communicates wirelessly with the base station computer via Omni-directional antennas.

One of the units was installed at a remote location near the perimeter of the base. **Figure 46** shows a picture of this unit, which will be referred to as node 1. The other unit, node 2, was installed on an observation tower overlooking an impact area. A picture of node 2 can be seen in **Figure 47**.

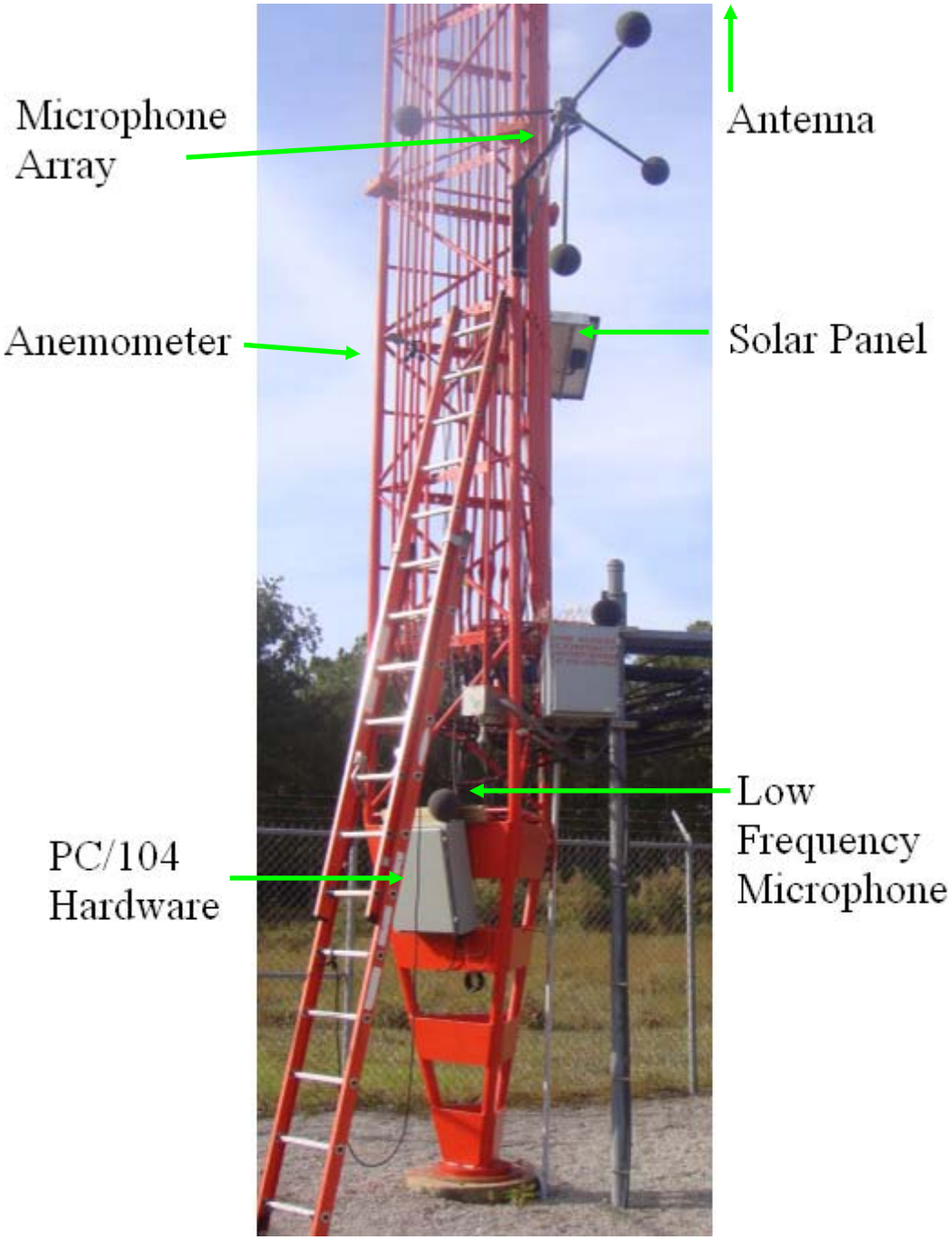

Figure 46. Picture of Node 1 Prototype Unit

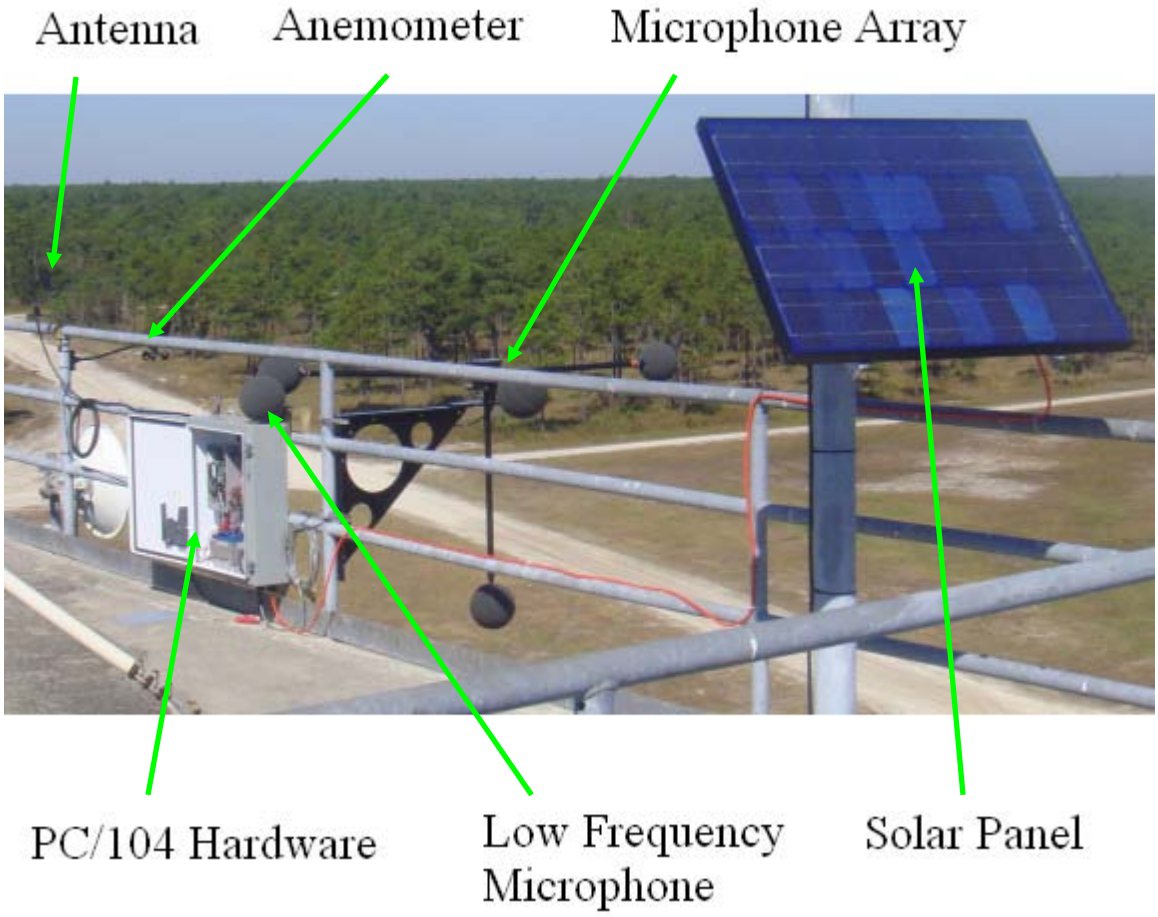

Figure 47. Picture of Node 2 Prototype Unit

6.3.2 Real Time Test Results

After the two prototype systems were installed and successfully debugged, the system was able to collect data automatically at each of the two prototype nodes. The BAMAS algorithm developed by APS runs continuously and activates whenever a given sound threshold is exceeded. For testing purposes, the threshold was set relatively low, at 90 dB. Once the system has been sufficiently tested, a higher threshold will be set as desired, most likely to 105 dB. Once the threshold is exceeded, the BAMAS algorithm

determines the likelihood of an acoustic event, using various methods, including methods similar to those described in section 5 of this thesis. If an acoustic event is found to be possible, the data are passed into the ANN classifier, otherwise the event is rejected as a false event. The BAMAS algorithm was found to successfully reject most waveforms consisting exclusively of wind. However, other non-impulsive events such as aircraft noise are able to pass through the BAMAS algorithm to the ANN classifier. Both false events and detection events are reported by each node to the base station, where they are logged to disk.

A data set of 90 waveforms was collected by the system during a testing period of approximately 30 minutes, during which live fire was taking place. For this testing period, all of the raw data were collected to verify the performance of the ANN classifier. A plot of the L_{pk} values versus time can be seen in **Figure 48**.

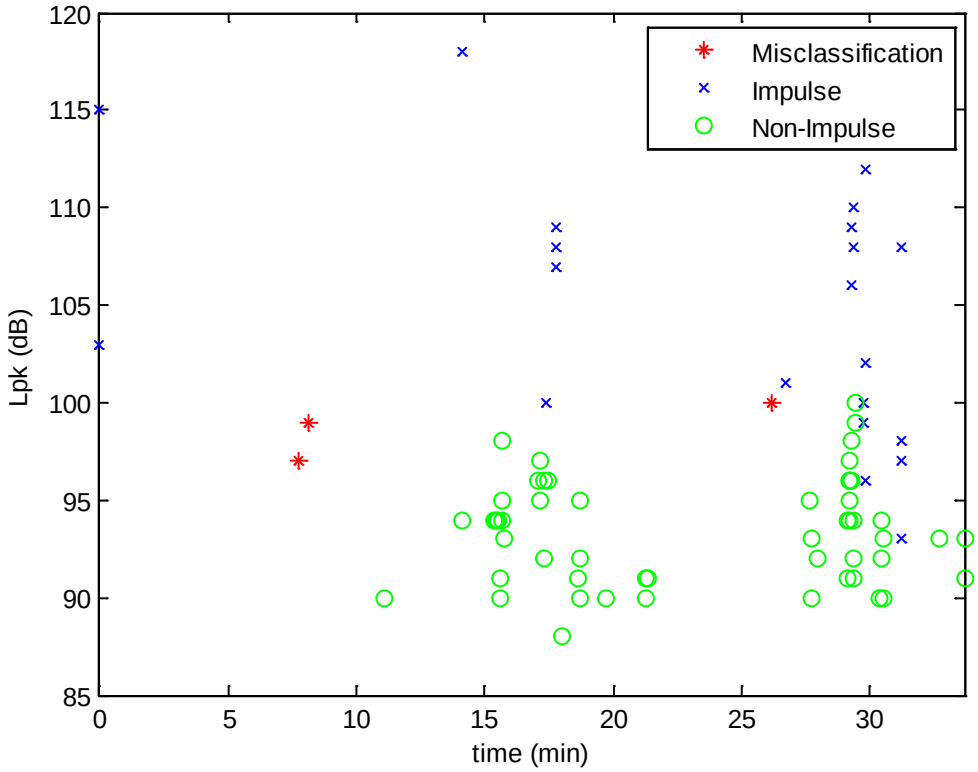

Figure 48. L_{pk} vs Time for Field Testing Results

Each of the waveforms was analyzed manually by looking at the waveform in both time and frequency domains, and also by listening to the waveform. In this manner the classification of each event was determined. These actual classifications were then compared with the ANN classifier output. Out of the 90 total waveforms, the ANN classifier misclassified three impulse waveforms as non-impulse. These three waveforms had relatively low peak levels, at 97, 99, and 100 dB. Plots of these three waveforms in both time and frequency domains can be seen in **Figure 49**, **Figure 50**, and **Figure 51**. It was determined that these events are not of great concern, due to their comparatively low L_{pk} values.

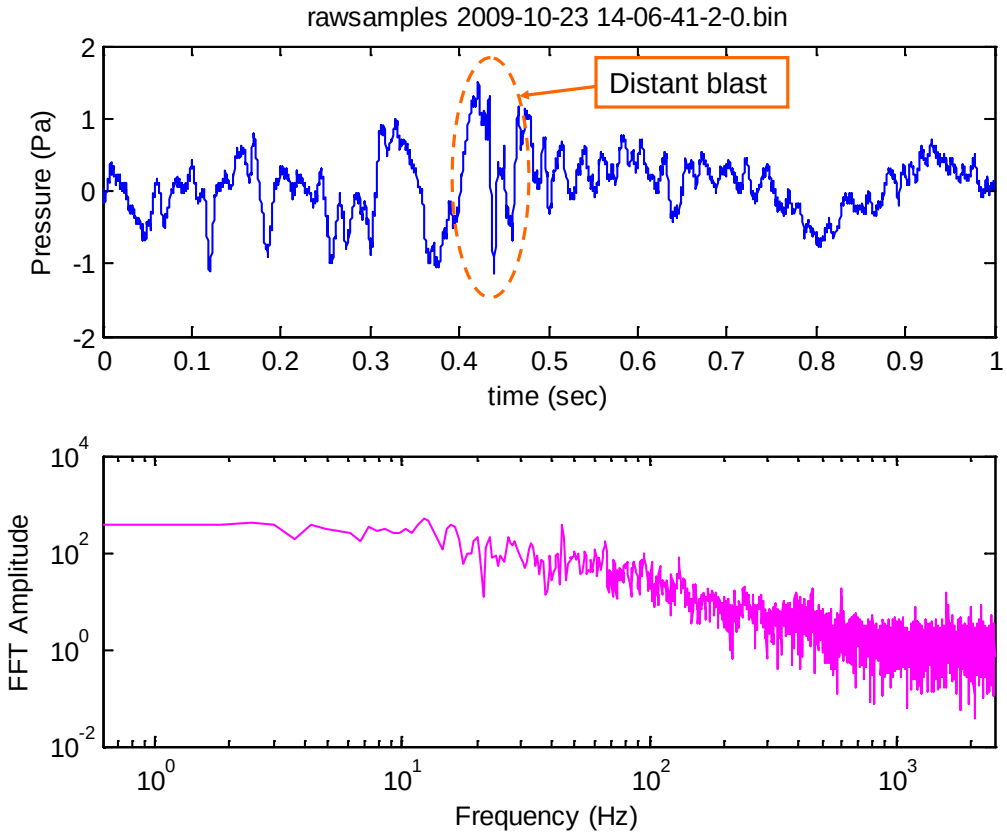

Figure 49. Plot of Misclassified Impulse Waveform 1

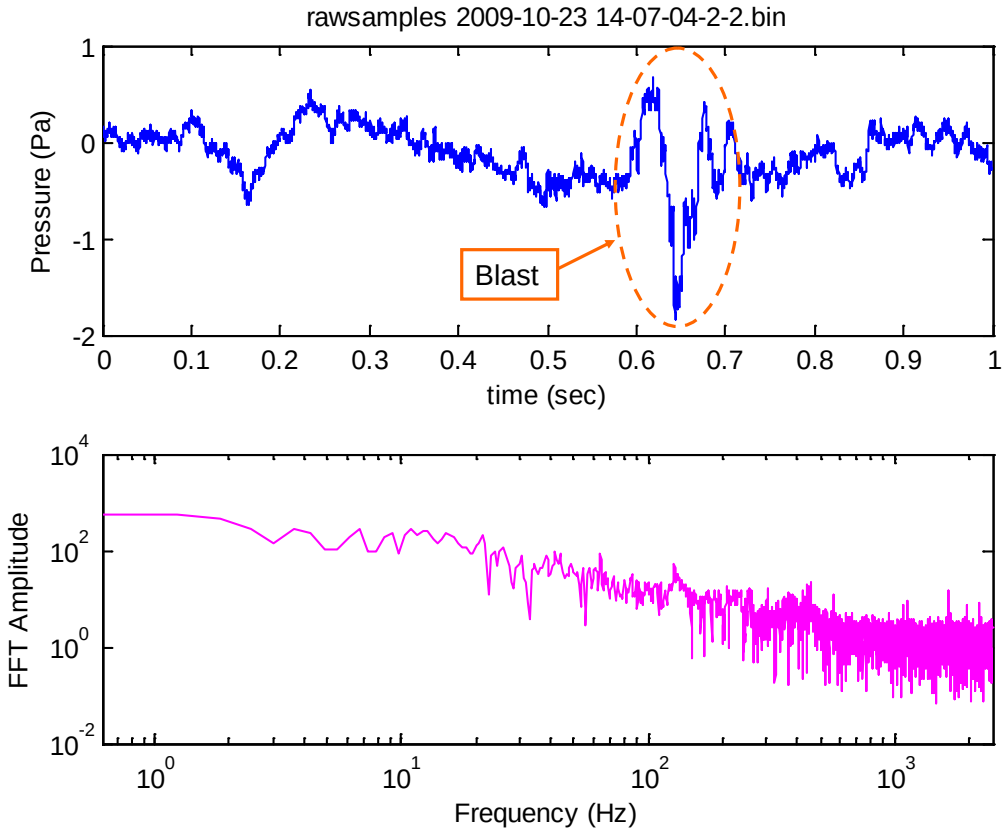

Figure 50. Plot of Misclassified Impulse Waveform 2

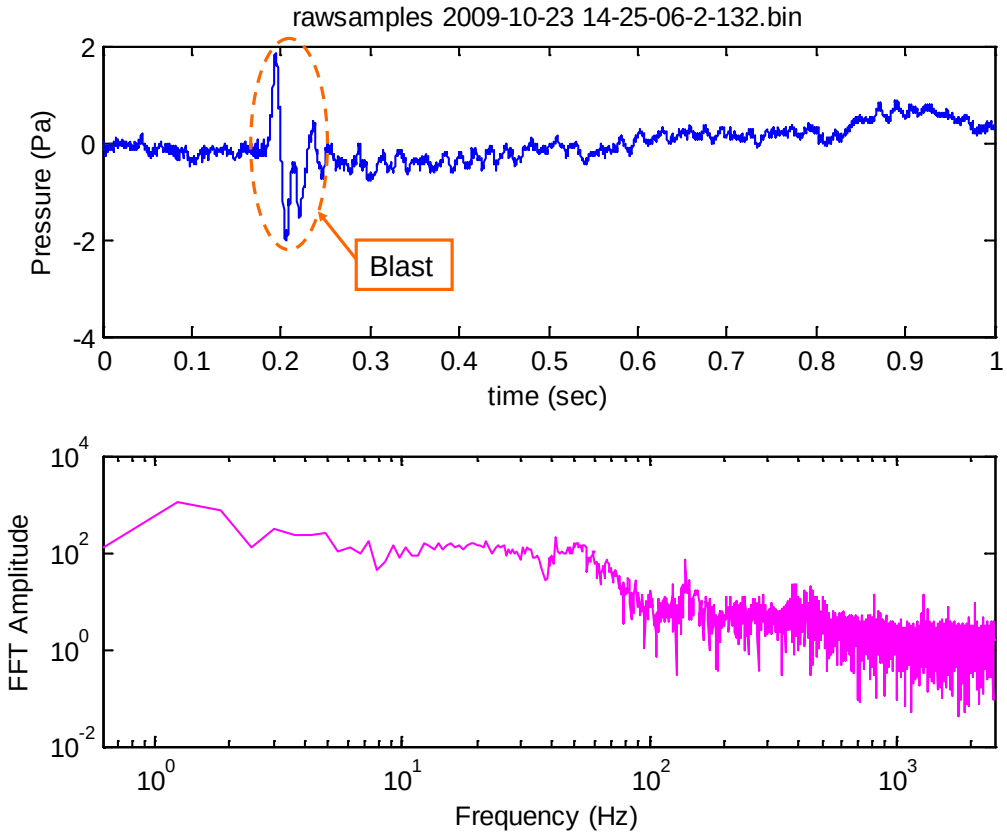

Figure 51. Plot of Misclassified Impulse Waveform 3

In addition to the 30 minute test window, the system continued to run and collect data. Due to restrictions on battery power, the raw data were not able to be collected in order to verify performance. Instead of looking at the raw waveforms directly, the results were analyzed by considering their calculated angle of incidence, classification result, and L_{pk} values. The angle of incidence, or bearing, is calculated by the APS BAMAS algorithm. Results were collected for a period of eight days from 10/29/2009 to 11/5/2009. The prototype systems remained active on these days from 8:00 – 18:00 EST. Only data from node 2 are considered, since node 1 did not collect significant amounts of

data. The L_{pk} values of all of the detected events at node 2 that were classified as impulse by the ANN classifier are plotted in **Figure 52**.

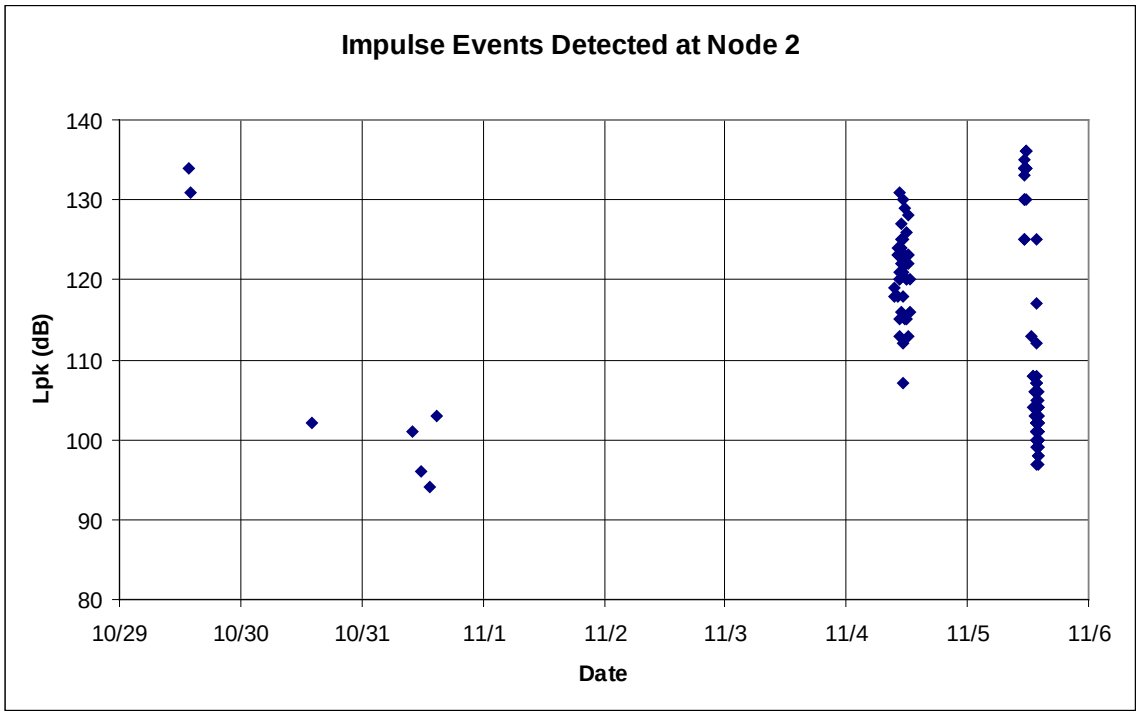

Figure 52. L_{pk} Values for Impulse Events Detected at Node 2

From **Figure 52** it can be seen that from 11/1/2009 to 11/3/2009 there were no detected impulse events. This is encouraging because based on the firing logs for those days, there were no large ordnance events taking place anywhere near node 2. Also, note that many impulse events were detected on 11/4/2009 and 11/5/2009. These events occurred across a relatively narrow window of time. Plots of the events from these dates can be seen in **Figure 53** and **Figure 54**.

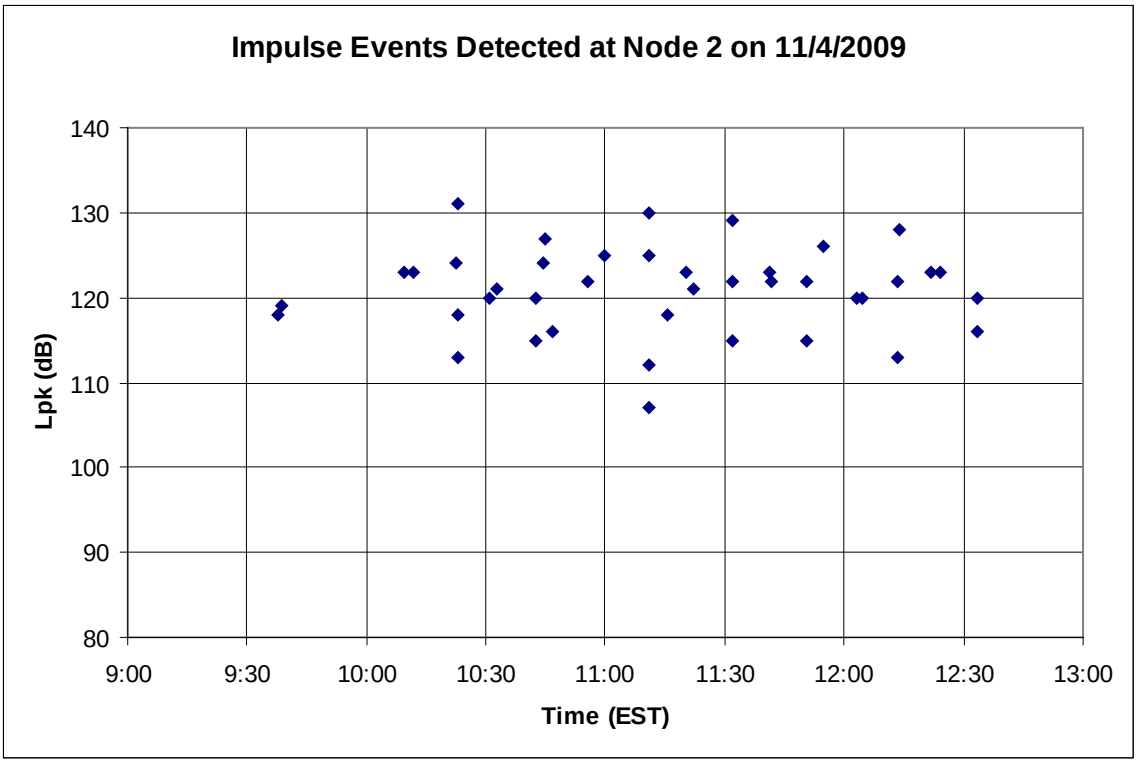

Figure 53. L_{pk} Values for Impulse Events Detected at Node 2 on 11/4/2009

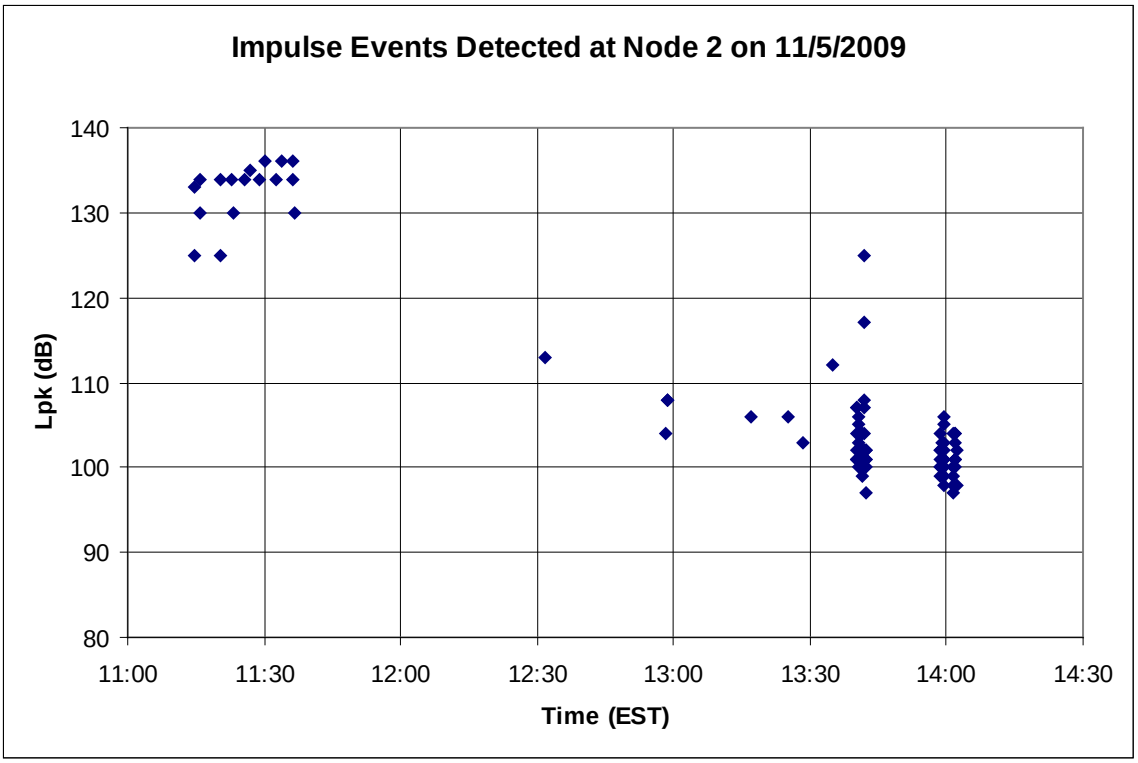

Figure 54. L_{pk} Values for Impulse Events Detected at Node 2 on 11/5/2009

Information about the location and time of impulse events was obtained from the military officials. Using this information, an approximate range of active areas was obtained for the given periods of time. These active areas are represented in **Figure 55** and **Figure 56** as a solid red region. The bearing of detected impulse events are also plotted in these figures as blue lines. **Figure 55** represents data from 11/4/2009, and **Figure 56** represents data from 11/5/2009.

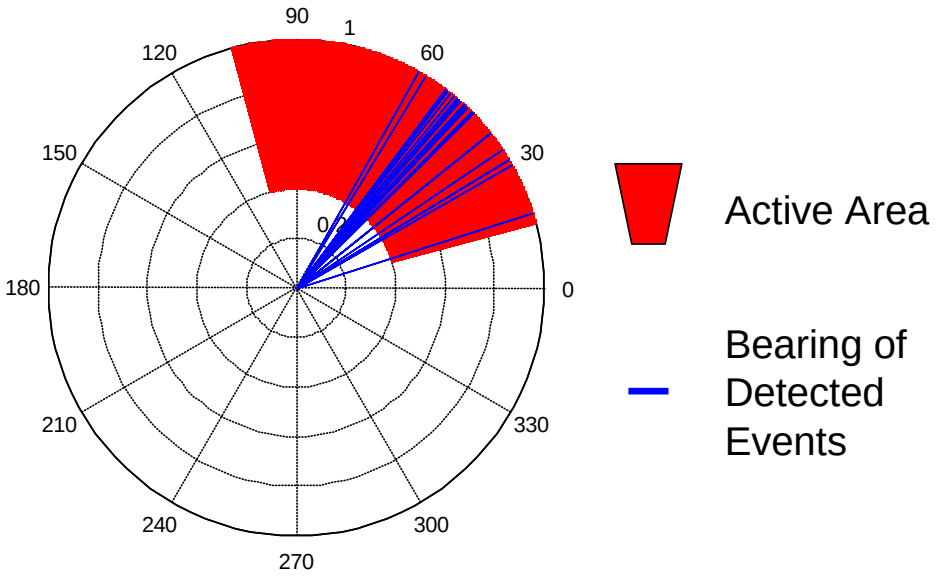

Figure 55. Calculated Bearing for Node 2 on 11/4/2009

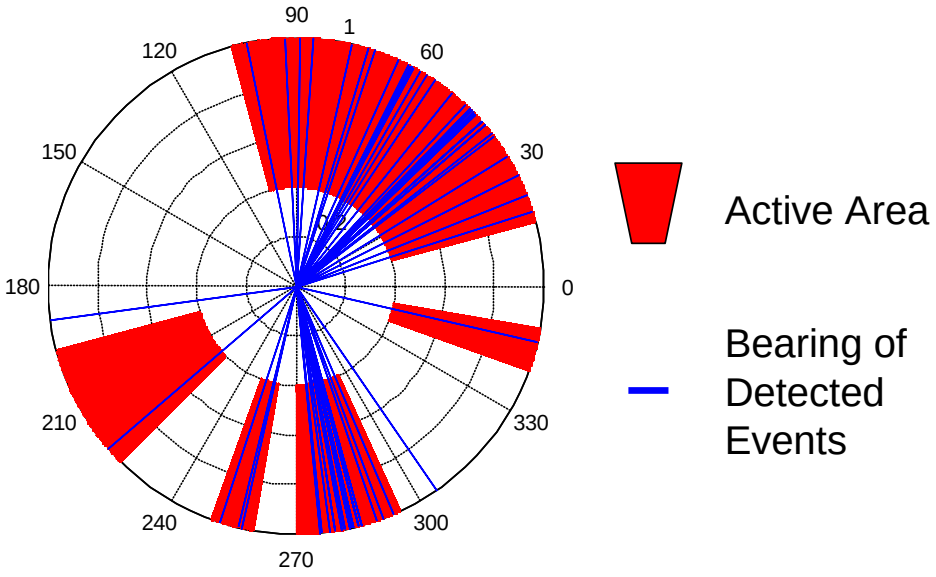

Figure 56. Calculated Bearing for Node 2 on 11/5/2009

From **Figure 55** and **Figure 56** it can be seen that the calculated bearing of each of the impulse events is approximately toward one of the active regions. Because of this, it is reasonable to assume that the classifier is properly identifying these events as impulse events. As seen in **Figure 56**, on 11/5/2009 the bearing of two events was not calculated to be in one of the active regions. For each of these events, however, the bearing is approximately toward one of these regions. This is still reasonable, since military training sometimes includes firing from different locations into the same region. It is also possible that the bearing is not calculated completely accurately, due to the precision of the fixed point computations that are involved.

7.0 CONCLUSIONS

A real time military impulse classifier has been developed for use in military installations. The overall system is able to successfully distinguish between impulsive events, such as artillery fire, and non-impulsive events, such as wind or aircraft noise. The system continually monitors peak sound levels via a microphone array, and reports any events that exceed a given L_{pk} threshold. An event which exceeds this threshold is analyzed using various array techniques, which determine whether or not the event is acoustic. If the event is determined to be acoustic, four scalar metrics are calculated from the data and passed into an artificial neural network (ANN) to determine whether the acoustic event is impulsive or non-impulsive.

The ANN classifier performance was evaluated individually for each base by using a single base's data exclusively as the testing data, and training and validating on data from all other bases. This process was repeated for each base to determine the generality of the classifier. Since the results yielded high accuracy for each base (> 99%), one overall ANN classifier can be applied globally to all bases. In order to maintain the generality of the classifier, data were also collected with various ordnance, weather, topography, and time of day. The final ANN was trained on over 2000 collected waveforms to 99.8% accuracy.

In addition to the previously developed single channel methods, multiple channel methods were investigated implementing a microphone array. These methods were

found to increase the effectiveness of properly classifying an event. Using the geometry of the microphone array, the bearing of an acoustic event can be calculated. This information was found to be useful in determining the source of an event. Additionally, wind noise can be rejected from the system due to the fact that it is not a propagating signal. A four channel microphone array was implemented in a prototype system created for field testing.

All of the collected data were sampled at 10 kHz with 16 bit resolution. In order to evaluate the fidelity of the data, the data were processed by a 400 Hz low pass 4^{th} order Butterworth filter, decimated from 16 bits to 12 bits, and re-sampled at 1 kHz. These processed data were used to train, validate, and test the ANN classifier. It was determined that the processed data yielded comparable results to that of the raw data for waveforms with L_{pk} values greater than 105 dB. Due to this result, hardware with less strict data fidelity requirements could be used.

In order to implement the impulse noise classifier in real time, the algorithm had to be coded for execution on a digital signal processing (DSP) board. The C programming language was selected for this purpose. The selected DSP board operates using a fixed point processor, therefore as many calculations as possible were coded using fixed point computation. However, some calculations required floating point computation, which was handled by the DSP board using fixed point emulation. This emulation process added to the execution time of the system, however the overall run time of the impulse classifier is approximately 0.25 seconds, which is sufficiently fast for real time applications.

Two prototype systems have been created and installed at a single military base in order to test and verify the results of the system in a real time application. Once the system was determined to be properly functioning, both raw data and classifier results were collected for a given test period. The results showed three misclassifications out of 90 events, none of which are of concern due to relatively low L_{pk} values. The classifier operated with 100% accuracy on events with L_{pk} values exceeding 100 dB. In addition, results were collected over a period of eight days and analyzed according to their classification result and calculated bearing. Using information about active locations on the base during this time period, detected impulse events were found to be originating approximately from one of these active areas.

8.0 FUTURE WORK

Further field testing at different locations would be useful for determining the robustness of the ANN classifier. Additionally, continuing to observe classifier performance during various weather conditions could provide information about the classifier accuracy as a function of temperature, barometric pressure, precipitation, *etc*. If performance is deemed insufficient, additional re-training can be performed on the network with newly collected data. The ANN could be re-trained globally for all bases. Alternatively, the ANN classifier could be re-trained specifically to each base if necessary or even specifically to each unit to obtain an accurate classification of the types of events that occur at a given location.

BIBLIOGRAPHY

Abraham, Bruce. "Impulse Noise Bearing and Amplitude Measurement and Analysis System." March 2005, Applied Physical Sciences Corporation. Available online at http://www.serdp.org/Research/upload/CP-1427.pdf

Anonymous. Office of Economic Adjustment, Office of Assistant Secretary of Defense, and Economic Security. "Joint Land Use Study." November 1993, Office of Economic Adjustment, DUSD(I&E), Suite 200, 400 Army Navy Drive, Arlington, VA 22202-2884, (703)604-6020.

ANSI S12.7-1986 (R1993). Methods for measurement of impulse noise. New York, NY: Acoustic Society of America.

Apex Embedded Systems STX104 Information. Available online at http://www.apexembeddedsystems.com/stx104.asp

Arcom PC104 PXA255 XScale Industrial Temperature Single Board Computer Information. Available online at http://www.arcom.com/pc104-xscale-viper.htm

J. Attias, A.Y. Duvdevany, I. Reshef-Haran, Michal Zilberg, and Nageris Beni. (2004) Military noise induced hearing loss. In: Handbook of effects of noise on man. Luxon, L (Ed), London.

J. Bendat and A. Piersol. *Random Data: Analysis and Measurement Procedures.* John Wiley and Sons, Inc. New York, NY, 2nd ed. 1986.

J. Bendat and A. Piersol. *Engineering Applications of Correlation and Spectral Analysis.* John Wiley and Sons, Inc. New York, NY, 2nd ed. 1993.

Jonathan W. Benson. "A real-time blast noise detection and wind rejection system." *Noise Control Engineering Journal*, **44** (6), November-December 1996, p. 306-314.

Robert D. Blevins. *Flow-Induced Vibration*, Krieger Publishing Co., Melbourne, FL, 2/e reprint, 2001.

Brian Bucci and Jeffrey Vipperman. "Artificial neural network military impulse noise classifier." IMECE2006-14065, Proceedings of IMECE 06: 2006 ASME International Mechanical Engineering Congress, November 5-10, 2006, Chicago, Illinois.

Brian Bucci and Jeffrey Vipperman. “Performance of artificial neural network classifiers to identify military impulse noise.” *Journal of the Acoustic Society of America*, **122** (3), September 2007 1602-1610.

Brian Bucci and Jeffrey Vipperman. “Bayesian military impulse noise classifier.” IMECE2007-41700, Proceedings of IMECE 07: 2007 ASME International Mechanical Engineering Congress, November 11-15, 2007, Seattle, Washington.

Brian Bucci and Jeffrey Vipperman. “Artificial neural network based temporal processing of waveforms to detect military impulse noise.” Proceedings of Noise Con 07: Institute of Noise Control Engineering Conference, October 22-24, 2007, Reno, Nevada.

B. Bucci. “Development of artificial neural network-based classifiers to identify military impulse noise.” University of Pittsburgh, Master’s Thesis, 2007.

B. Bucci and J. Vipperman. “Bayesian military impulse noise classifier.” IMECE2007-41700, Proceedings of IMECE 07: 2007 ASME International Mechanical Engineering Congress, November 11-15, 2007, Seattle, Washington.

B. Bucci and J. Vipperman. “An investigation of the characteristics of a bayesian military impulse noise classifier.” Proceedings of NoiseCon 2008/ASME NCAD, July 28-30, 2008, Dearborn Michigan.

E. Buchta and J. Vos. “A field survey on the annoyance caused by sounds from large firearms and road traffic.” *Journal of the Acoustic Society of America*, **104**(5), November 1998 2890-2902.

Edwin Chong and Stanislaw Zak, *An Introduction to Optimization.* John Wiley and Sons, Inc. New York, NY, 2nd ed. 2001.

Igor Chunchuzov, Sergey Kulichov, Alexander Otrezov, and Vitaly Perepelkin. “Acoustic pulse propagation through a fluctuating stably stratified atmospheric boundary layer.” J. Acoustic Society of America, 117 (4), April 2005, p. 1868-1879.

A. Cichocki and R. Unbehauen. *Neural Networks for Optimization and Signal Processing*. John Wiley and Sons Ltd., New York, NY, 1993.

Malcolm J. Crocker. *Handbook of Acoustics*. John Wiley and Sons, Inc. New York, NY, 1998.

John Eargle. *The Microphone Book*. Focal Press, Burlington, MA, 2nd ed. 2004.

Fast Artificial Neural Network Library (FANN). Available online at http://leenissen.dk/fann/

Daniel Graupe. *Principles of Artificial Neural Networks*, World Scientific Publishing Co. Pte. Ltd., Singapore, 1997.

Simon Haykin. *Array Signal Processing*, Prentice Hall, 1985.

Simon Haykin. *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, 1999.

Don H. Johnson and Dan E. Dudgeon. *Array Signal Processing: Concepts and Techniques*. Prentice Hall, Englewood Cliffs, NJ, 1993.

George A. Luz, Edward T. Nykaza, Catherine M. Stewart, Larry L. Pater. The role of sleep disturbance in predicting community response to the noise of heavy weapons. Final Report Work Unit CNN-T352, US Army Corps of Engineers Engineer Research and Development Center, ERDC/CERL TR-04-26, November 2004.

George A. Luz. "Suggested procedures for recording noise complaints at army installations." United States Army Environmental Hygiene Agency, Aberdeen Proving Ground, MD 21010-5422. HSE-08/WP Technical Guide, April 1, 1980.

I.A. McCowan. "Robust Speech Recognition using Microphone Arrays," PhD Thesis, Queensland University of Technology, Australia, 2001.

Warren McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." *The Bulletin of Mathematical Biophysics*, Vol. 5, 1943.

Michael Norton, Denis Karczub. *Fundamentals of Noise and Vibration Analysis for Engineers*, Cambridge University Press, New York, NY, 2nd ed. 2003.

S. Morgan and R Raspet: Investigation of the mechanisms of low-frequency wind noise generation outdoors, J. Acoust. Soc. Am., **92**(2), 1180/1183(1992).

Press, William H. *Numerical Recipes in C: The art of scientific computing*. Cambridge University Press, New York, NY, 2nd ed. 1997.

Singiresu S. Rao. *Mechanical Vibrations*. Pearson Education, Inc., Upper Saddle River, NJ, 4th ed. 2004.

Matthew Rhudy, Brian Bucci, Jeffrey Vipperman, Jeffrey Allanach, and Bruce Abraham. "Microphone Array Analysis Methods Using Cross-Correlations." IMECE2009-10798, Proceedings of IMECE 09: 2009 ASME International Mechanical Engineering Congress, November 13-19, 2009, Lake Buena Vista, Florida.

Paul Schomer and Keith Attenborough, "Basic results from full-scale tests at Fort Drum", *Noise Control Engineering Journal*, **53** (3), May-June 2005, p. 94-109.

Paul D. Schomer and Robert D. Neathammer, *Community Reaction to Impulsive Noise: A Final 10-Year Research Summary* (U.S. Army Construction Engineering Research Laboratories Technical Report, N-167, June 1985).

Paul D. Schomer, Aaron J. Averbuch, and Lester M. Lendrum, *An Army Blast Noise Warning and Monitoring System* (U.S. Army Construction Engineering Research Laboratories Technical Report, N-88/03, February 1988).

Tri-M Engineering HESC104 60 Watt High Efficiency UPS PC104 Information. Available online at http://www.tri-m.com/products/engineering/hesc104.html

Sergiy A. Vorobyov, Alex B. Gershman, and Zhi-Quan Luo. *IEEE Transactions on Signal Processing*, **51**(2), February, 2003 313-324.

Source Forge. Available online at http://sourceforge.net/projects/kissfft/.

WxUSA U.S. Weather Reports. Available online at http://www.wxusa.com/.